Copyright
by
Murari Mani
2008

The Dissertation Committee for Murari Mani Certifies that this is the approved version of the following dissertation:

Robust Algorithms for Area and Power Optimization of Digital Integrated Circuits under Variability

Committee:

Michael Orshansky, Supervisor

Adnan Aziz

David Morton

David Pan

Constantine Caramanis

Robust Algorithms for Area and Power Optimization of Digital Integrated Circuits under Variability

by

Murari Mani, B.E.; 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
December, 2008

To my parents

Acknowledgements

I would like to express my heartfelt thanks to my advisor Prof. Michael Orshansky for his guidance during the past five years. I owe a considerable portion of my technical and research acumen to his tutelage, and will remain forever indebted. This dissertation would not have been possible without him. It was a pleasure working with Michael and it is with a gleam in my eye that I turn to rebuff the skeptics who contend that a Ph.D. is 'not fun'.

I am extremely grateful to the other members on my committee, Prof. Adnan Aziz, Prof. David Morton, Prof. David Pan, and Prof. Constantine Caramanis for their helpful comments and suggestions. My thanks to Prof. Morton for helping us to formulate the statistical gate sizing problem, and Prof. Caramanis, whose expertise was indispensable to the adaptable buffer project.

I was fortunate to be accorded the opportunity to apply our algorithms at AMD and I would like to thank my manager Mahesh Sharma for giving me this chance. I would also like to express my gratitude to my collaborators Ashish Singh, Ku He and Anirudh Devgan, my lab mates Wei-Shen Wang, Bin Zhang, Shayak Banerjee and Hady Zeineddine and my colleagues at AMD Yaping Zhan and Jeegar Shah for many an insightful discussion.

I made many great friends during my time at UT Austin and they have given me many memories to cherish. A special nod of appreciation goes to Sriram and Sankar for never declining an invitation to grab a cup of coffee and to Anand for livening up the dull days with his sarcasm.

I count myself very lucky to be an integral part of the lives of remarkable people such as my parents and my grandmother. Words are seldom adequate and often unnecessary and I am not going to try to express the gratitude I feel towards them. Sincere thanks are due to my friends Kaushik and Prem for putting up with me for eight years, and my wife Manju for her love and support. Finally, a big thank you to *kutti* for all the cheer and joy over the years.

Robust Algorithms for Area and Power Optimization of Digital Integrated Circuits under Variability

Publication No._____________

Murari Mani, Ph.D.
The University of Texas at Austin, 2008

Supervisor: Michael Orshansky

As device geometries shrink, variability of process parameters becomes pronounced, resulting in a significant impact on the power and timing performance of integrated circuits. Deterministic optimization algorithms for power and area lack capabilities for handling uncertainty, and may lead to over-conservative solutions. As a result, there is an increasing need for statistical algorithms that can take into account the probabilistic nature of process parameters. Statistical optimization techniques however suffer from the limitation of high computational complexity. The objective of this work is to develop efficient algorithms for optimization of area and power under process variability while guaranteeing high yield. The first half of the dissertation focuses on two design-time techniques: (i) a gate sizing approach for area minimization under timing variability; (ii) an algorithm for total power minimization considering variability in timing and power.

Design-time methods impose an overhead on each instance of the fabricated chip since they lack the ability to react to the actual conditions on the chip. In the second half of the dissertation we develop joint design-time and post-silicon co-optimization techniques which are superior to design-time only optimization methods. Specifically, we develop (i) a methodology for optimization of leakage power using design-time sizing and post silicon tuning using adaptive body bias; (ii) an optimization technique to minimize the total power of a buffer chain while considering the finite nature of adaptability afforded. The developed algorithms effectively improve the over-conservatism of the corner-based deterministic algorithms and permit us to target a specified yield level accurately. As the magnitude of variability increases, it is expected that statistical algorithms will become increasingly important in future technology generations.

Table of Contents

TABLE OF CONTENTS IX

List of Tables xii

List of Figures xiii

Chapter 1: Introduction 1
- 1.1 Statistical Optimization For Timing Yield 3
- 1.2 The Need For Power Limited Parametric Yield Optimization 5
- 1.3 Design-Time and Post-Silicon Co-optimization: Motivation and Challenges 10
- 1.4 Design Of Power-Optimal Buffers Tunable to Process Variability 12
- 1.5 Dissertation Organization 13

Chapter 2: A Fast Sizing Algorithm by Robust Linear Programming 15
- 2.1 Formulating the Robust Linear Program For Sizing 16
  - 2.1.1 Deterministic Problem Formulation 16
  - 2.1.2 Sizing under Parameter Uncertainty 16
- 2.2 Statistical gate delay modeling 17
- 2.3 Path Based Sizing Using Second Order Conic Programming 20
  - 2.3.1 Overview of Second Order Conic Programming 20
  - 2.3.1 Formulation of Path–based Constraints 21
  - 2.3.2 Node- based Sizing Formulation 22
- 2.4 Criticality Based Yield Assignment 22
- 2.5 Application Of The Algorithm In The Microprocessor Design Flow 30
- 2.6 Experimental Results 32
- 2.7 Summary 36

Chapter 3: A Statistical Algorithm for Power- & Timing-Limited Parametric Yield Optimization 37
- 3.1 Power Minimization by Delay Budgeting 38
- 3.2 Statistical Model for Power-Delay Sensitivity 42

3.3 Delay Budgeting Using Robust Linear Programming 45
3.4 Statistical Delay Budgeting Using SOCP 47
    3.3.1 Transforming the Circuit Timing Constraint 47
    3.3.1 Handling the Probabilistic Power Constraint.............................48
3.5 Experimental Results 50
    3.4.1 Effectiveness of Algorithm in Optimizing Power 52
    3.4.2 Computational Properties of the Algorithm...............................56
3.6 A Case Study on The c17 Benchmark 56
3.7 Summary 59
Chapter 4: Joint Design-Time and Post-Silicon Minimization of Parametric Yield Loss using Adjustable Robust Optimization..60
4.1 Gate And Circuit Modeling 61
    4.1.1 Delay and Leakage Models..61
    4.1.2 Affine Model for Body Bias 64
4.2 Design Time / Post Silicon Co-Optimization Using Adaptable Robust Optimization 65
    4.2.1 Adaptable Robust Optimization...66
    4.2.2 Co-Optimization: Problem Formulation 67
4.3 Problem Solution 69
    4.3.1 Accounting for Intra-chip Variation 71
4.4 Experimental Results 72
4.5 Summary 78
Chapter 5: Design of Power-Optimal Buffers Tunable to Process Variability......79
5.1 Adaptable Buffer Design Principles 80
    5.1.1 Overview of Finite Adaptable Optimization..............................80
    5.1.2 Buffer Design as a Finite Adaptable Optimization Problem 81
    5.1.3 Design of Adaptable Buffer 82
5.2 Buffer Optimization Strategy..83
    5.2.1 Partitioning the Uncertainty Set...83
    5.2.2 Formulating the Constraints and Objective 84
5.3 Problem Solution Using Geometric Programming 86

5.3.1 Extension to Variability in Threshold Voltage 87
5.4 Experimental Results 87
5.5 Summary 90

Chapter 6: Conclusions 91

Bibliography 94

Vita ……………………………………………………………………….....100

List of Tables

Table 2.1: Results of non-uniform assignment on circuit with 100 inverter chains 27

Table 2.2: Impact of logic depth, number of paths and correlation on area savings 28

Table 2.3: Results of yield assignment on benchmark circuits.. 30

Table 2.4: Minimum area obtained by deterministic and statistical algorithms at different yield levels. 35

Table 3.1: Delay and Power characteristics of low-V_{th} and high-V_{th} devices 39

Table 3.2: Power Savings Obtained by Statistical over Deterministic Algorithm 55

Table 4.1: Leakage power savings obtained by the joint post-silicon and design-time optimization. 77

List of Figures

Figure 1.1: Predicted variability of key parameters, circuit timing, and power consumption (Data source: ITRS) 2

Figure 1.2: Exponential dependence of leakage current on 0.18μm process parameters results in a large spread for relatively small variations around their nominal value 6

Figure 1.3: Inverse correlation between leakage power and frequency contributes to parametric yield loss.. 8

Figure 1.4: Effectiveness of ABB in reducing delay spread.......... 9

Figure 2.1: The piecewise linear delay model for a NAND2 gate.......... 18

Figure 2.2: Equiprobability contours of jointly distributed normal random variables. 20

Figure 2.3:The yield assignment flow 25

Figure 2.4: Simple circuit for evaluating yield assignment strategy 26

Figure 2.5: Uniform node yield selection strategy on a benchmark circuit.......... 30

Figure 2.6: The pseudo-code for application of statistical sizing to a sequential circuit.. 31

Figure 2.7: The area-delay curves for different breakdowns of variability. Statistical optimization results in smaller area for the same delay target as the intra-chip component increases 33

Figure 2.8: The run-time behavior of the sizing algorithm. 34

Figure 2.9: Result of performing Monte Carlo on circuit configurations for different desired yield levels for the c6288 benchmark.......... 34

Figure 2.10: The area-delay curves at different yield levels. Statistical optimization does uniformly better than the deterministic optimization at the same yield level. 35

Figure 3.1: The power-delay space for a NAND2 gate driving two different capacitive loads. The Pareto frontier is depicted by the dashed gray lines. 40

Figure 3.2: Flowchart illustrating the algorithm for power minimization. 42

Figure 3.3: PDFs of static (leakage) power produced by a Monte- Carlo simulation of the benchmark circuit (C432) optimized by the deterministic and statistical algorithms. 51

Figure 3.4: Power-delay curves for 99.9% timing and power yield. 52

Figure 3.5: Power-delay curves at different timing yield for the C432 benchmark……..53
Figure 3.6: Power-delay curves at different power limited yields.................................... 54
Figure 3.7: Run time behavior of the statistical power optimization algorithm 55
Figure 3.8: c17 circuit sized for maximum slack... 57
Figure 3.9: c17 circuit configuration produced by deterministic optimization 57
Figure 3.10: c17 circuit configuration produced by statistical optimization 58
Figure 4.1: The dependence of delay on body bias. 62
Figure 4.2: Comparison of the normalized leakage of inverter predicted by SPICE and the analytical leakage model... 63
Figure 4.3: The 2-phase algorithm optim_abb for post silicon optimization using ABB..71
Figure 4.4: PDFs of delay distributions produced by design- time only optimization and joint optimization.. 72
Figure 4.5: Comparison of disjoint optimization and joint design time and post silicon optimization. 73
Figure 4.6: Comparison of design time only optimization and joint design time and post silicon optimization.. 74
Figure 4.7: The runtime increases as the measurement complexity is increased, as optimal policy depends on more measurements. 75
Figure 4.8: The expected value of leakage power decreases as we increase measurement complexity but the benefits level off for high values of *k*. 76
Figure 4.9: The expected value of leakage power decreases as we increase control complexity *n*... 77
Figure 5.1: Conceptual representation of the adaptable buffer....................................... 81
Figure 5.2: Adaptable buffer design using tristate inverters.. 83
Figure 5.3: Pseudo-code for solving the adaptive buffer design problem 86
Figure 5.4: Dependence of total power of buffer configuration truncation point for different timing yields... 88
Figure 5.5: Comparison of statistical and adaptive buffer design approaches 89
Figure 5.6: Monte Carlo simulation of buffer configuration 89

Chapter 1: Introduction

The growth of process variability in scaled CMOS requires that it is explicitly addressed in the design of high performance and low power microprocessors. This growth can be attributed to multiple factors, including the difficulty of manufacturing control, the emergence of new systematic variation-generating mechanisms, and the increase in fundamental atomic-scale randomness – for example, the random placement of dopant atoms in the transistor channel. The International Technology Roadmap for Semiconductors (ITRS) in 2006 indicates that the variability of the parameters could be as much as 30% of the nominal value, resulting in 40% variation in circuit timing and 50% variation in power dissipation for the current generation (65nm technology) [1]. Moreover, it is anticipated that variability in parameters will continue to increase according to the current trend, as shown in Figure 1.1. This growth of process variability has ushered in an urgent need for statistical analysis and optimization algorithms

Recently, considerable research efforts have focused on developing statistical approaches to timing analysis, including the models and algorithms accounting for the impact of delay variability on circuit performance [2][3][4] [5]. These techniques concern themselves with eliminating the conservatism introduced by employing traditional worst-case timing models in predicting the timing yield of the circuit. In view of the importance of variability, new methods are needed to evaluate the power-limited parametric yield of integrated circuits and guide the design towards statistically feasible and preferable solutions. This can be achieved through the migration to statistical optimization techniques that account for both power and delay variability.

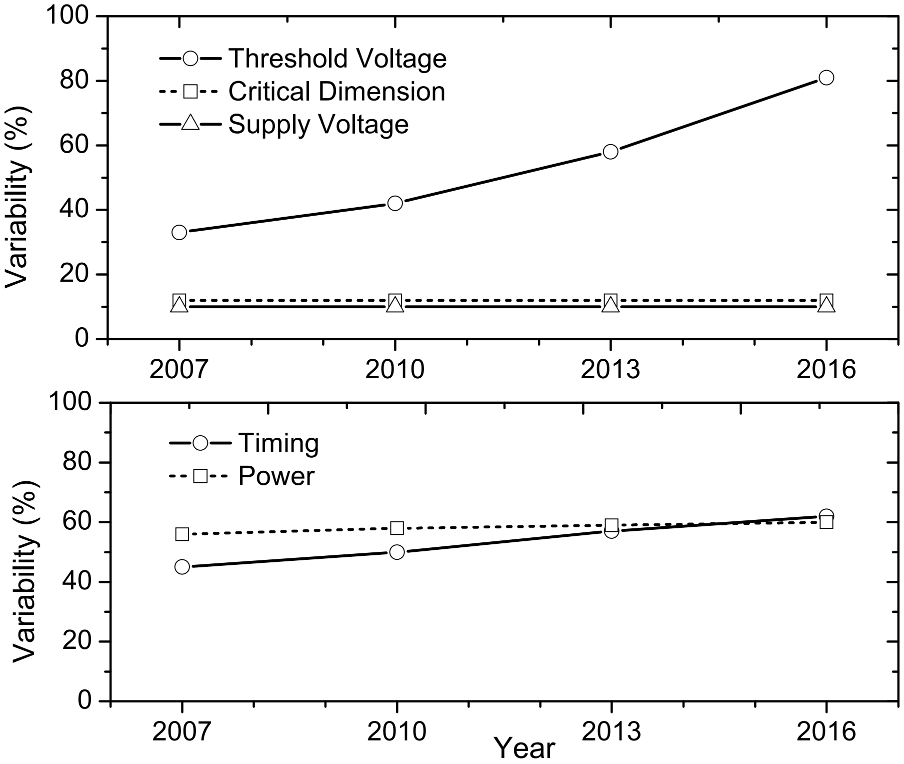


Figure 1.1: Predicted variability of key parameters, circuit timing, and power consumption (Data source: ITRS)

Several factors contribute to the growth in process variability [6][7][8]. While the continued need for more performance necessitates rapid technology scaling, there are severe limitations to our capacity to improve manufacturing tolerances [9]. This is manifested in the rise of such effects as channel length variation due to the optical proximity effect [10][11]; systematic spatial gate length variation due to the aberrations in the stepper lens [12]; and variation in interconnect properties caused by non-uniform rate of chemical-mechanical polishing (CMP) in layout regions of different pattern density[13][14]. Scaling also brings about parameter uncertainty of a fundamental atomic-level nature. This is best exemplified by variability in transistor threshold voltage due to random dopant fluctuations (RDF). As transistors scale, the transistor channel contains fewer dopant atoms whose precise number and location cannot be controlled, and even small fluctuations can impact threshold voltage significantly [15][16] [17].

The patterns of variability are also changing: the intra-chip component of variation grows as a percentage of total variability in key process parameters such as channel length and threshold voltage [18][19]. It is this change that is largely responsible for the need to develop new approaches to timing analysis and optimization, as the traditional methods fail in the presence of uncorrelated intra-chip variability.

Circuit-level variability is directly dependent on the decision variables: for instance, the standard deviation of threshold voltage depends inversely on the square root of transistor area [20]. Statistical algorithms that explicitly account for the variance of objective and constraint functions during optimization are expected to perform much better. In contrast, deterministic algorithms lack the notion of parameter variance and parametric yield, preventing design for yield as an active design strategy. An algorithm that does not comprehend the dynamic changes in performance variability arising from threshold voltage dependency on sizing is unlikely to be successful in parametric yield optimization. Instead, if a worst case process corner is assumed to ensure sufficient yield the circuit gets over-designed resulting in worse power consumption and lower performance. Thus, the introduction of statistical optimization has a potentially significant impact on circuit performance and parametric yield.

1.1 STATISTICAL OPTIMIZATION FOR TIMING YIELD

Traditional circuit optimization techniques are insufficient for the purpose of parametric yield improvement in nanometer scale integrated circuits. In the past, case-files have been used effectively with the traditional deterministic algorithms while guaranteeing a specific yield point. Typically, these case files would be worst case, nominal, and best case process corners combined with the worst case, nominal, and best case operating (voltage, temperature) corners. The effect of variability was captured in these case files by modifying the device SPICE model parameters to correspond to a specific percentile of the parameter distribution [21]. Analyzing and optimizing the circuit with these parameters guaranteed that it would meet the performance constraints at a specific percentile of probability. However, this approach works only when variability is predominantly inter-chip, causing differences in the chip-to-chip properties, with parameter variation in devices within a chip being neglected. In nanometer scale technologies, intra-chip variation is significant. Also, deterministic optimization makes the tacit assumption that circuit performances of different gates have identical sensitivities to the variation of process parameters. The highly non-linear and non-additive responses of performance variability make this premise untenable [22]. This results in the breakdown of the case-file based approach to handling variability in optimization as it becomes impossible to come up with a case file that will guarantee a specific yield point.

The inability to target a specific parametric yield point is a significant limitation of deterministic optimization, in general, and sizing, one of the most potent optimization techniques, in particular. The transistor and gate sizing problem has been formulated in several ways in the deterministic setting, including unconstrained delay minimization [23], and area and power minimization under delay constraints [24]. Several powerful solution methods exist, among which is sensitivity-based iterative approach of TILOS [25], an optimization approach using linear programming [26] and a fast technique based on Lagrangian relaxation [27] However, none of these approaches take variability into account, treating gate delays as fixed quantities.

There have been several recent attempts to introduce statistical considerations into sizing [28][29][30][31][32]. In [28], a general non-linear gate sizing problem based on a statistical gate delay model is proposed. An approach based on the concepts from utility theory [29] posits an objective function that penalizes paths with large variance. In [30], the authors modify the existing sizing algorithm based on Lagrangian relaxation [27] to incorporate an additional yield constraint Recently, two techniques based on geometric programming have been proposed [31][32] and solved using convex optimization tools.

In this dissertation, we present a new approach to statistical gate sizing. The problem is cast into a robust linear program, which is then reformulated as a second-order conic program to analytically capture the dependence of the objective function on the variance of gate delays in closed form. This allows us to achieve better run-time compared to the known approaches.

1.2 THE NEED FOR POWER LIMITED PARAMETRIC YIELD OPTIMIZATION

The increase in leakage power with scaling, and the strong dependence of leakage on highly varying process parameters, raises the importance of statistical leakage and parametric yield modeling. There are several reasons for increased leakage power consumption. Supply voltage scaling requires the reduction in threshold voltage (V_{th}) in order to maintain gate overdrive strength. Threshold voltage reduction causes an exponential increase in subthreshold channel leakage current. To make matters worse, aggressive scaling of gate oxide thickness leads to significant gate oxide tunneling current [33].

For transistors in the weak inversion region, the subthreshold current can be expressed as:

$$I_{sub} \propto e^{(V_{gs}-V_{th})/\eta V_{thermal}}\left(1-e^{-V_{ds}/V_{thermal}}\right) \tag{1.1}$$

where V_{gs} and V_{ds} are gate- and drain-to-source voltages, $V_{thermal}$ is the thermal voltage, and η is the subthreshold slope coefficient [33]. For the purpose of statistical analysis, the exponential dependence of subthreshold current on process parameters is better captured by an empirical model in terms of the variation in effective channel length (ΔL) and the variation in threshold voltage (ΔV), taken to be stochastically independent of channel length [34]:

$$I_{sub} \propto e^{-(\Delta L + a_2 \Delta L^2 + a_3 \Delta V)/a_1} \tag{1.2}$$

where a_1, a_2 and a_3 are process-dependent parameters. The gate tunneling current strongly depends on the thickness of oxide (T_{ox}) and can be described as [35]:

$$I_{ox} \propto e^{(c_1 V_{gs} - c_2 T_{ox}^{-2.5})} + e^{(c_1 V_{gd} - c_2 T_{ox}^{-2.5})} \tag{1.3}$$

where c_1 and c_2 are the process-dependent fitting parameters, and V_{gs} and V_{ds} are the gate-to-source and gate-to-drain voltages respectively. A simple empirical model captures the dependence of I_{ox} on the variation in the oxide thickness (ΔT) [34]:

$$I_{ox} \propto e^{-\Delta T/b} \tag{1.4}$$

where b is the process-dependent parameter.

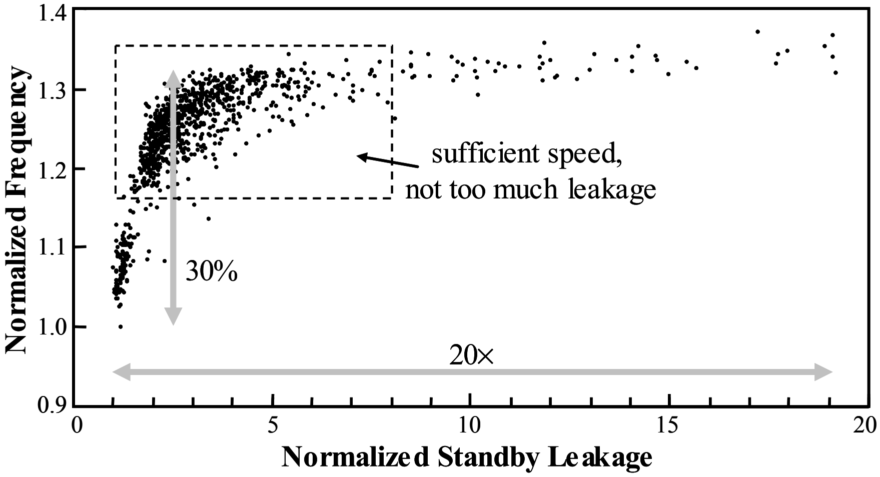


Figure 1.2: Exponential dependence of leakage current on 0.18µm process parameters results in a large spread for relatively small variations around their nominal value

The models indicate that both subthreshold and gate leakage currents are exponential functions of highly-variable process parameters, specifically effective channel length, threshold voltage, and oxide thickness. This strong dependence causes a large spread in leakage current in the presence of process variations (Figure 1.2), with subthreshold leakage depending primarily on L_{eff} and V_{th}, and gate leakage depending on $T_{ox.}$ Historically, T_{ox} has been a well-controlled parameter, and as a result, it had smaller impact on leakage variability. However, this is rapidly changing as technology approaches the limits of thin film scaling. While leakage power exhibits exponential dependencies on process variables, chip frequency has a near-linear dependency on most parameters [34]. This difference in magnitude of variation is easily observed in measurements. Figure 1.2 shows that a 1.3× variation in delay between fast and slow die could potentially lead to a 20× variation in leakage current [36].

Leakage power is inversely correlated with chip frequency. Slow die have low leakage, while fast die have high leakage (Figure 1.2). The same parameters that reduce gate delay – shorter channel length, lower threshold voltage, thinner gate oxide – also increase the leakage. Moreover, the spread in leakage grows as the chip becomes faster. In characterizing chips according to their operating frequency, it has been observed that a substantial portion of the chips in the fast bins have unacceptably high leakage power consumption.

In the absence of substantial leakage power, parametric yield is determined by the maximum possible clock frequency. Switching power is relatively insensitive to process variation. When the leakage power typical of current CMOS technologies is added, the total power starts approaching the power limit determined by the cooling and packaging considerations. Crucially, the exponential dependence of leakage on process spread means that the total power may cross the cooling (power) limit well below the maximum

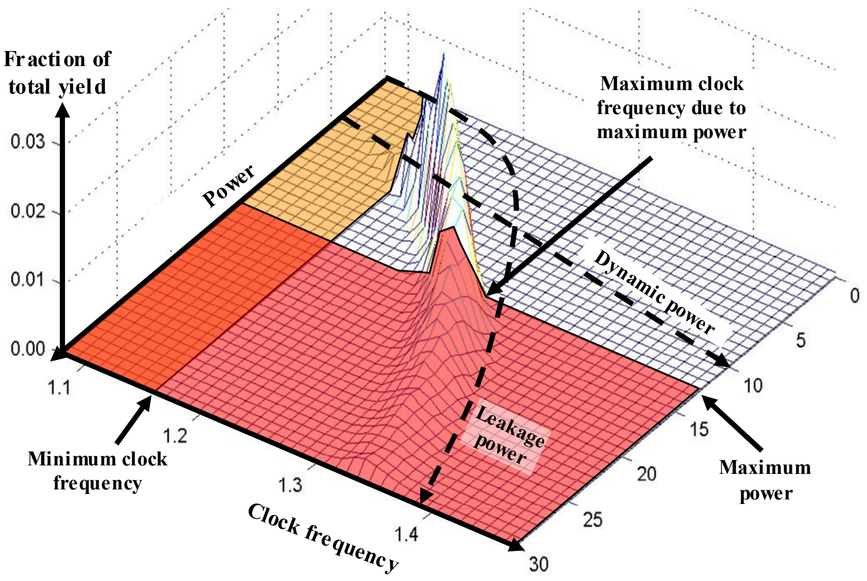


Figure 1.3: Inverse correlation between leakage power and frequency contributes to parametric yield loss. The maximum frequency of usable chips is reduced because chips in what would be the "fast" bin exceed power limits.

possible chip frequency, since chips operating at higher frequencies have exponentially higher leakage power consumption. Thus, due to the inverse correlation between speed and leakage, yield is limited both by slower chips and chips that are too fast, because they are too leaky.

This is further illustrated in Figure 1.3. The leakage-delay correlation and the resulting dual squeeze on parametric yield is one of the reasons why new methods that can simultaneously estimate timing-limited and power-limited yield need to be utilized. These will allow designing circuits in a way that optimizes the trade-off between power- and timing-limited yields. At the same time, this will permit making the circuit more robust, i.e., less sensitive to parameter variation. For this, new methods that can simultaneously handle timing-limited and power-limited yield need to be utilized. This requires new techniques for statistical parametric yield optimization.

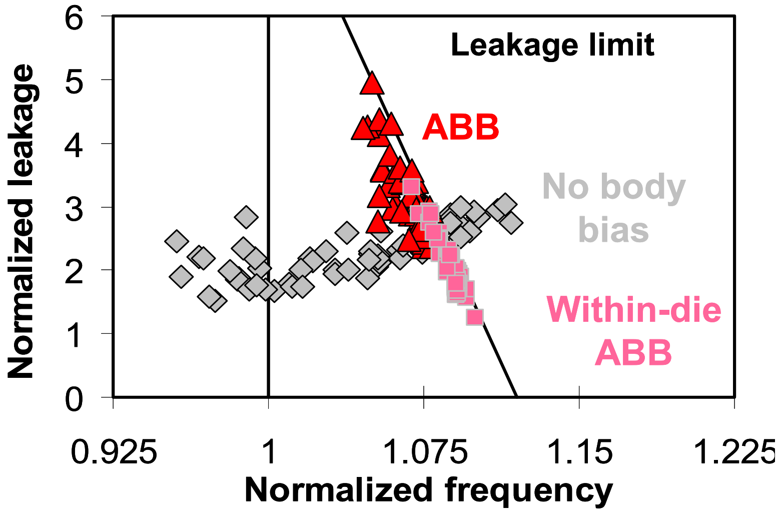


Figure 1.4: Effectiveness of ABB in reducing delay spread

In this dissertation, we describe a new statistical algorithm for total power minimization. Starting with a chance-constrained LP, it is transformed into a second order conic program using mathematical properties of the uncertain parameters. A two phase approach based on optimal delay budgeting and slack utilization, akin to [37] is used. The delay budgeting phase is formulated as a robust version of the power-weighted linear program that assigns slacks based on power-delay sensitivities of gates. We explicitly incorporate the notion of variability in delay and power due to process variations into the optimization, by setting an uncertain robust linear program. The variance of delay and power, assumed to be due to channel length and threshold voltage variation, is mapped to the variance of the sensitivity vector. The statistical (robust) linear program is cast into a second order conic program that can be solved efficiently. The slack assignment is interleaved with the configuration selection which optimally redistributes slack to the gates in the circuit to minimize total power savings.

1.3 DESIGN-TIME AND POST-SILICON CO-OPTIMIZATION: MOTIVATION AND CHALLENGES

Two fundamental paradigms are available for dealing with variability: statistical design (optimization at design time) and post-silicon adaptivity (on-line tuning). To guarantee reliable circuit operation with minimal power consumption, next-generation circuit synthesis techniques for robustness must explicitly account for the availability of post-silicon adaptivity in synthesizing the circuit. There is a growing body of work on statistical circuit analysis methods and statistical post-synthesis optimization, including sizing and dual-threshold voltage assignment algorithms. These tools show promise in reducing parametric yield loss, or alternatively, reducing power consumption while maintaining high yield. However, the growing magnitude and complexity of uncertainty is bound to make post-synthesis tuning techniques insufficient in guaranteeing reliable circuit operation with reasonable parametric yield.

Post-silicon design adaptivity, or tuning, currently includes several techniques; the primary ones are adaptive body biasing (ABB) and adaptive supply voltage (ASV). ABB uses the body effect to modulate the threshold voltages of transistors, thereby controlling leakage and performance [38][39][40][41] (Figure 1.4). ASV raises the power supply (V_{dd}) for slow (low-leakage) dies, and lowers it for fast (high-leakage) dies, ensuring better overall yield [42]. It relies on the roughly cubic dependence of leakage power on Vdd in CMOS circuits (also impacting dynamic power quadratically).

A widespread industrial adoption of adaptive techniques is not yet possible for two reasons. One is that designers do not have the tools to help them decide whether, and how much, adaptive circuitry is needed, or what type of post-silicon tuning technique will be most appropriate. The availability of both design-time (pre-silicon) optimization and post-silicon adaptivity leads to a rich optimization space in which coordination between the two levels is required. Sizing can be used to upsize the gates beyond the need of a nominal design to achieve higher timing yield, but with increased power. Alternatively, the adaptivity of threshold voltage can be used to tighten the speed distribution to improve yield. Depending on the magnitude and the spatial structure of variability, the two approaches will have different cost-effectiveness, i.e., they will be characterized by different Pareto curves in the space of design objectives.

Algorithmically, future robust circuit synthesis can be conceptualized as a two-stage optimization problem, with additional second-stage tuning available upon the realization of uncertain variables. In this dissertation an efficient formulation is proposed using the theory of adjustable optimization. This optimization paradigm presumes that the decision-maker has a chance to update his optimization strategy upon learning additional information. If the objective function is linear in the decision variables, then, under the conditions that the uncertainty sets are affine functions of some parameters, the optimal policy for the second-stage decisions can be computed efficiently.

The problem is formulated in the following way. The first-stage (design-time) power-delay optimization is done via sizing, and second-stage (post-silicon) optimization is achieved by body bias tuning. The second stage decision variables are represented as affine function of parameter uncertainty. The solution to this optimization problem is a design time decision (size of gates in the circuit) and an optimal policy that prescribes the amount of bias depending on the realizations of uncertain variables (e.g. gate length, V_{th} on a specific chip). Three measures of complexity that parameterize the solution and the optimality of this problem are introduced by us: the control complexity (the granularity of control), the measurement complexity (the granularity of the monitoring and sensing circuitry), and the parameter complexity (a measure of how spatially uncorrelated the process variable is). Using these metrics, formal quantitative trade-offs between design-time and post-silicon adaptivity can be identified. Such capability will also be useful for the analysis and development of the fine-granular control structures, e.g. for determining the spatial granularity.

1.4. DESIGN OF POWER-OPTIMAL BUFFERS TUNABLE TO PROCESS VARIABILITY

Large capacitive loads are ubiquitous in CMOS integrated circuits. Typically, tapered buffers are designed to drive these large capacitances to ensure that the load placed on previous stages of the signal path is not too large [43]. Buffers are used in memory access path as wordline drivers [44], to drive large off-chip capacitances in I/O circuits [45] and in clock trees to ensure that skew constraints are satisfied [46]. Also, the recent trend of exacerbating wire delays necessitates the insertion of more buffers per unit length of global interconnect to meet delay targets [47]. Aggressive deployment of buffers in high-performance microprocessors means that they now account for a significant portion of total power consumption of the chip. For instance, wordline drivers are estimated to account for nearly half the energy consumption of small embedded SRAMs [45].

The expedient need for power efficiency in mobile and portable devices, in conjunction with the increase in leakage power with scaling, has espoused the development of techniques for low-power buffer design [48][49]. With the growth of variability, several techniques have been proposed for statistical power optimization in general, and buffer design in particular [50][51], to reduce parametric yield loss due to variability. However design-time methods impose a fixed overhead for each instance of the fabricated chip. An alternate paradigm to design-time optimization is post-silicon adaptivity, which allows the designer to tune chips individually to help meet performance constraints.

One specific methodology for run-time adaptivity for buffer chains explored in [51] in the context of memories, is to use the capability of switching between high-speed and low-power configurations to exploit their delay-energy tradeoffs. The alternative power-delay characteristics can be achieved using different techniques like sizing, or by employing different threshold voltages. However, the strategies developed thus far do not take into account the magnitude and characteristics of process variability to design the buffers. Using our approach, we demonstrate that the optimal decision depends on the underlying process variability and propose a method to co-ordinate the design-time and post-silicon steps optimally and efficiently.

We propose a general formulation for tunable buffer design under uncertainty using the theory of finite adaptable optimization. Under this framework, the buffers are designed to have either an additional branch. The best configuration is selected based on the realization of uncertainty. The optimization problem is formulated as the minimization of the total power consumption while guaranteeing that timing yield constraints are met. The solution to the problem is the set of design time decisions, namely the sizing of the inverters in the buffers, and the optimal tuning policy. The tuning policy of selecting an alternative design after manufacture can be described by defining an *optimal partition of the uncertainty set*: the partitions are the regions in the space of process parameters. A decision about the buffer configuration option is taken depending on the region into which a particular realization falls.

1.5 DISSERTATION ORGANIZATION

The remainder of the dissertation is organized as follows. Chapter 2 presents a statistical sizing algorithm which is formulated as a robust linear program. In Chapter 3, the impact of variability on power is discussed and an algorithm for total power minimization is presented which treats both timing and power probabilistically. Power reduction is performed by simultaneous sizing and dual threshold voltage assignment, and good run-time is achieved by casting the problem as a second-order conic problem. In Chapter 4 we develop an algorithm for design-time and post-silicon co-optimization which unifies design-time gate-level sizing and post-silicon adaptation using adaptive body bias at the chip level. The formulation utilizes adjustable robust linear programming to derive the optimal policy for assigning body bias once the uncertain variables, such as gate length and threshold voltage, are known. In Chapter 5 we develop a strategy to optimize the total power of a buffer circuit in the presence of variability by designing a tunable buffer circuit wherein depending on realizations of process parameters, buffer stages with different size are selected. The number of alternative buffer tunable settings is small, and we show power can be reduced by choosing an optimal rule that guides switching between the alternatives once the uncertain process parameters are realized. Finally, Chapter 6 concludes this dissertation.

Chapter 2: A Fast Sizing Algorithm by Robust Linear Programming

Statistical gate sizing has emerged as an important tool to reduce the over-conservatism of deterministic corner-case based optimization while ensuring that timing yield constraints are satisfied. It is possible to formulate a general statistical gate sizing problem that can be described by analytical but non-linear functions and solve it directly using a general non-linear solver [28]. The objective and constraints are expressed as explicit functions of the mean and variance of gate delays. However, the techniques relying on non-linear optimization tend to be excessively slow which would greatly limit the capacity for large-scale circuit optimization. More efficient formulations based on geometric programming are also possible. In [31], the fact that sizing problems have fairly flat maxima is exploited by utilizing heuristic techniques to compute the "soft-max" of arrival times. Statistical static timing analysis is then used to guide the optimization in the right direction. The algorithm based on geometric programming presented in [32] models parameter variations using an uncertainty ellipsoid, and proceeds to construct a robust geometric program, which is solved by convex optimization tools.

In this chapter, we present a new approach to statistical gate sizing. Its major contribution is the analytical treatment of delay variability and an efficient computation implementation. The problem is cast into a robust linear program, which is then reformulated as a second-order conic program to analytically capture the dependence of the objective function on the variance of gate delays in closed form. Second-order conic programs can be very efficiently solved using existing interior point methods. This allows us to achieve significantly better run-time compared to the known approaches. We demonstrate the use of the sizing algorithm on an industrial microprocessor module and a number of practical challenges of using a statistical algorithm in an industrial setting are addressed. The variability and delay models are generated from and validated by industrial technology files and transistor models and the algorithm was integrated into a CAD flow handling sequential elements, fixed size macros, and non-static logic elements.

2.1 FORMULATING THE ROBUST LINEAR PROGRAM FOR SIZING

This section first presents the mathematical formulation of deterministic circuit sizing. Then it proposes a robust linear program for sizing in the presence of variability to guarantee the attainment of the desired timing yield.

2.1.1 Deterministic Problem Formulation

The gate sizing problem can be formulated as follows. Given a circuit implemented using standard library cells and the maximum delay target, find a load drive capability for all the cells that meets timing constrains while minimizing total circuit area. We assume that the cell's drive capabilities are continuous within some range. Both the drive strength and the input capacitance of cells are known once the sizing parameter is determined. The problem can be written as:

$$\begin{aligned} &\min \sum_j c_j s_j \\ &s.t.\ \ D_{ckt} \le D_{target} \end{aligned} \tag{2.1}$$

where, s_j is the size of gate j and c_j is a weight assigned to each gate type representing the area of its minimum sized version, D_{ckt} is the delay of the circuit and D_{target} is the timing target.

2.1.2 Sizing under Parameter Uncertainty

The impact of process variability is to introduce uncertainty into circuit timing. Several process parameters exhibit an impact on the timing performance of high-end integrated circuits. Notably, the variability in the effective channel length, L_{eff}, and the threshold voltage, V_{th}, cause substantial variation in gate delay. Intra-chip component of variation that introduces uncorrelated delay variation has particularly detrimental effect on timing. In seeking a formulation of the statistical sizing problem, we must satisfy two requirements. First, the probabilistic constraints have to be represented in a way that will permit analytical treatment of circuit timing variance. Second, this analytical formulation must be amenable to efficient computational implementation.

We can write the statistical equivalent of the deterministic problem (2.1) by making the constraint satisfaction a probability event. This results in the following robust linear program for sizing:

$$\begin{aligned} &\min \sum_j c_j s_j \\ &s.t.\ \ P(D_{ckt} \le D_{target}) \ge \gamma \end{aligned} \tag{2.2}$$

Now we require the constraints to be met with probability of γ, which is the required timing yield at the timing target D_{target}.

2.2 STATISTICAL GATE DELAY MODELING

Following earlier suggestions, a piece-wise linear approximation can be used in optimization [26]. We adopt a linearized gate delay modeling method, extending it to a statistical representation. Let the gate delay be represented as $d_j = \overline{d}_j + \Delta d_j$, where $\overline{d}_j$ is the nominal gate delay and Δd_j is the term representing the variability in delay. The nominal gate delay can be described by the piecewise linear equations:

$$\overline{d}_j = a_{j1}^l - a_{j2}^l s_j + a_{j3}^l \sum s_k \ \forall\, l \in [1, L] \tag{2.3}$$

where l are the fitting regions and L is the number of such regions. This model captures the dependence of delay on the size of the gate s_j and its load $\sum s_k$. The coefficients found are by the least-square fit of the piecewise linear model to a set of data points

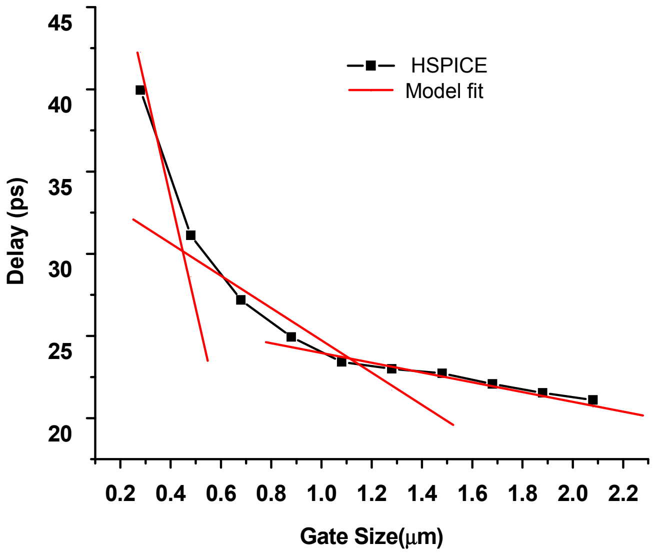


Figure 2.1: The piecewise linear delay model for a NAND2 gate.

generated via a circuit simulation using SPICE for gates in the cell library. The accuracy of the fit can be improved by increasing the number of fitting regions L. The accuracy of the approximation is good and the average error is less than 5% for $L=3$. The size range considered is 1-8x of the minimum size gate.

Assuming that a first order Taylor's series expansion for gate delay d_j is adequate, we can write:

$$\Delta d_j \cong (\partial d_j / \partial L)\Delta L + (\partial d_j / \partial V_{th})\Delta V_{th} + (\partial d_j / \partial W)\Delta W \qquad (2.4)$$

where ΔL, ΔV_{th} and ΔW are the parameter random from nominal. The precise dependence of the sensitivities, i.e. the first derivatives of delay with respect to the parameter, on gate size is posynomial. To capture the dependence of the variance of gate delay on the decision variables (gate sizes) of a second-order conic program, the sensitivity coefficients of a gate need to be represented as linear functions of the driver and load sizes. We use an empirically fitted linear model for this purpose. For example, the sensitivity of delay to gate length variation is empirically modeled as:

$$(\partial d_j / \partial L) = (b_{j0} + b_{j1} s_j + b_{j2} \sum s_k) \tag{2.5}$$

Also, to simplify analysis, we force the sensitivity coefficients to be the same in the entire size range. The values of sensitivities of delay to gate length, width, and threshold voltage are computed by SPICE simulations. The error in mean is less than 2% while the error at the $\mu + 3\sigma$ point is around 8%. Note that since L_{eff}, V_{th} and W can be modeled as normally distributed random variables, gate delay is also normal.

The modeling framework must consistently handle different decompositions of variability into inter-chip and intra-chip components. This is accomplished here by adopting a linear additive model that decomposes the variability of all parameters into the intra-chip and chip-to-chip variability components. For example, for the effective channel length the model is: $\Delta L = \Delta L_{inter} + \Delta L_{intra}$, and $\sigma_L{}^2 = \sigma_{inter}{}^2 + \sigma_{intra}{}^2$. The gate delay co-variance can be written down as:

$$\operatorname{cov}(d_j, d_k) = \frac{\partial d_j}{\partial L}\frac{\partial d_k}{\partial L}\operatorname{cov}(L_j, L_k) + \frac{\partial d_j}{\partial W}\frac{\partial d_k}{\partial W}\operatorname{cov}(W_j, W_k) + \frac{\partial d_j}{\partial V_{th}}\frac{\partial d_k}{\partial V_{th}}\operatorname{cov}(V_{th_j}, V_{th_k}) \tag{2.6}$$

where, $\operatorname{cov}(L_j, L_k) = \operatorname{cov}(\Delta L_{inter} + \Delta L^j_{intra}, \Delta L_{inter} + \Delta L^k_{intra})$ and ΔL_{inter}, $\Delta V_{th_{inter}}$ and ΔW_{inter} are the inter-chip components and ΔL_{intra}, $\Delta V_{th_{intra}}$ and ΔW_{intra} are the intra-chip components of variation. This model is not based on the knowledge of the specifics of the spatial correlation of intra-chip variability, in contrast to approaches such as [2]. Thus, the gate-to-gate correlation is assumed to come from the joint impact of intra- and inter-chip variability only. We believe this is a good model to sufficiently approximate the percent point function of gate delay (that gives the value of delay at an arbitrary percentile of the distribution) as a second-order cone of gate sizes, which was our initial objective in developing the statistical gate delay model that will be suitable for statistical optimization.

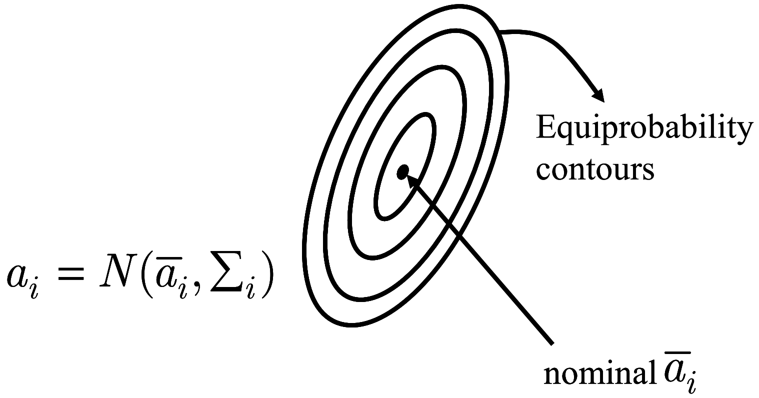


Figure 2.2: Equiprobability contours of jointly distributed normal random variables.

2.3 PATH BASED SIZING USING SECOND ORDER CONIC PROGRAMMING

In this section the theory behind Second Order Conic Programming (SOCP) is briefly explained. We then proceed to transform the robust LP in (2.2) into a path based formulation which is an SOCP.

2.3.1 Overview of Second Order Conic Programming

In general, a Second Order Conic Program (SOCP) consists of minimizing a linear function over the convex set described by the intersection of an affine space with one or more second-order cones. Consider the Robust Linear Program:

$$\begin{aligned}&\min c^T x\\ &s.t.\ P(a_i^T x \le b_i) \ge \eta_i,\ i = 1..m\end{aligned} \tag{2.7}$$

where $a_i = N(\overline{a}_i, \Sigma_i)$.

The equi-probability Gaussian sets are ellipsoids with their axis and orientation given by covariance matrix Σ_i (Figure 2.2). This implies that the constraint $P(a_i^T x \le b_i) \ge \eta_i$ can be written as:

$$b_i - \overline{a}_i^T x \ge \phi^{-1}(\eta_i) \left\| \Sigma_i^{1/2} x \right\| \tag{2.8}$$

This leads to the following SOCP equivalent of the robust LP:

$$\begin{aligned} &\min c^T x \\ &\bar{a}_i^T x + \phi^{-1}(\eta_i)(x^T \Sigma_i x)^{1/2} \le b_i \end{aligned} \tag{2.9}$$

The reasons for seeking an SOCP based formulation for sizing are manifold. Second-order conic programs are convex, and an optimal solution is, therefore, globally optimal. In addition there exist extremely efficient techniques to solve SOCPs that exploit their special structure [53][54].

2.3.1 Formulation of Path–based Constraints

The probabilistic constraint on circuit delay must be translated into a set of path-based constraints, in the form $P(D_i \le D_{target}) \ge \alpha_i$, where D_i is the delay of path *i,* such that the resulting set of constraints well approximates, and ideally guarantees, the specified yield level γ. The path delay is:

$$D_i = \sum d_j,\ \forall j \in p_i \tag{2.10}$$

Given the models of the previous section, path delays are Gaussian, $D_i \sim N(\bar{D}_i, \sigma_{D_i}^2)$, and using translation-invariance of normal distribution, the probabilistic path delay constraint $P(D_i \le D_{target}) \ge \alpha_i$ can be re-written as:

$$P((D_i - \bar{D}_i) / \sigma_{D_i}) \le (D_{target} - \bar{D}_i) / \sigma_{D_i}) \ge \alpha_i \tag{2.11}$$

which can be finally transformed into:

$$\bar{D}_i + \phi^{-1}(\alpha_i)\sigma_{D_i} \le D_{target} \tag{2.12}$$

where $\bar{D}_i$ is the nominal delay of the path *i,* σ_{D_i} is the standard deviation of the delay of the path, and ϕ is the cumulative distribution function (*cdf*) of the standard normal distribution $N(0,1)$. The path delay variance can now be expressed as a second-order conic function of gate sizes (s_j) using the delay covariance expressions established in the previous section:

$$\sigma_{D_i}(s_j) = \left(\sum_{j \in p_i} \sum_{k \in p_i} \operatorname{cov}(d_j, d_k) \right)^{1/2} \tag{2.13}$$

This permits setting up the SOCP for statistical gate sizing:

$$\begin{aligned} &\min \sum_j c_j s_j \\ &s.t.\ \overline{D}_i + \phi^{-1}(\alpha_i)\sigma_{D_i} \leq D_{target},\ \forall i \in P \end{aligned} \tag{2.14}$$

2.3.2 Node- based Sizing Formulation

Since the above constraints are path based, the resulting optimization problem is computationally expensive. Converting the path delay constraints into node-based constraints, in mathematical terms, requires additively approximating the percent-point function of path delay with a combination of node-delay percent point functions. The transformation is performed using the standard introduction of additional node arrival time constraints:

$$\begin{aligned} &\min \sum_j c_j s_j \\ &AT_o \leq D_{target},\quad \text{for } \forall o \in PO \\ &AT_k \geq AT_j + \overline{d}_j + \phi^{-1}(\beta_j)\sigma_{d_j} \end{aligned} \tag{2.15}$$

where AT_j is the arrival time at node j, D_{target} is the required arrival time at the primary outputs. Here $\overline{d}_j$ and $\sigma_{d_j} = \operatorname{cov}(d_j, d_j)^{1/2}$ are the mean and standard deviation of the gate delay. The transformation to the node-based formulation, involved selecting the node probability levels (β_j) for the individual gates on the path. In the following section we discuss a strategy for node yield assignment which is based on path criticality.

2.4 CRITICALITY BASED YIELD ASSIGNMENT

In the previous section, we outlined the statistical sizing algorithm assuming that the node yields β_j are chosen in such a way that required timing yield γ is obtained and the objective is minimized over all such assignments. While such an assignment is not intuitively obvious, it is clear that an exhaustive enumeration of all possible assignments is infeasible. However, it is apparent that a successful yield assignment scheme has to integrate information about the circuit obtained from statistical timing. In this section we derive a strategy based on path and gate criticalities that guides the optimization in the direction of optimality. Let us initially proceed from a path based setting, which is more intuitive.

Definition 1: The criticality of a path i is defined as [55] $\zeta = \Pr(D_i > D_k)$ $\forall k \in P, k \neq i$.

Definition 2: The criticality of a path can also be expressed in terms of the sensitivity of the circuit delay to the change in path delay as $\zeta_i = \partial E[D_{ckt}] / \partial E[D_i]$.

Fact 1: The criticalities of all the paths in a circuit sum to 1.

Definition 2 allows us quantify the impact of a change in path yield $\Delta\alpha_i$ on circuit yield γ as:

$$\Delta\gamma = \zeta_i \Delta\alpha_i \frac{(\partial F_{ckt} / \partial D_{ckt})}{(\partial F_i / \partial D_i)} \tag{2.16}$$

where F_{ckt} and F_i are the *cdf*s of the circuit and path delay respectively.

The first order derivatives of the *cdf*s with respect to delay represents the sensitivity of change in yield to the corresponding delay. The above equation expresses the change in the overall circuit yield in terms of the change in the individual path yields. Observing that criticality of path determines its impact on overall circuit yield, we conclude that more critical paths should be assigned larger path yields. This leads to the following LP for path yield assignment:

$$\begin{aligned} &\max \sum \Delta\alpha_i \\ &s.t. \sum \zeta_i \Delta\alpha_i \frac{(\partial F_{ckt} / \partial D_{ckt})}{(\partial F_i / \partial D_i)} \leq \Delta\gamma_{iter} \\ &\alpha_i = \alpha_i^{init} - \Delta\alpha_i \\ &\alpha_i \geq \gamma_{target}, \ \Delta\alpha_i \geq 0 \end{aligned} \tag{2.17}$$

The goal is to maximize the relaxation in path-yield $\Delta\alpha_i$ from the initial value α_i^{init} such that the yield target γ is met. The criticalities ζ_i , initial circuit (γ_{init})and path yields (α_i^{init}) are obtained from Statistical Static Timing Analysis (SSTA). The procedure is repeated iteratively by moving in small decrements of $\Delta\gamma_{iter}$ to reach γ .

The transformation to a node-based formulation, involves selecting the node probability levels (β_j) for the individual gates on the path. To achieve this we employ a similar criticality based strategy to formulate an LP which gives us the node based relaxations $\Delta\beta_j$:

$$\begin{aligned}
&\max \sum \Delta\beta_j \\
&s.t. \sum c_i \Delta\alpha_i \frac{(\partial F_{ckt} / \partial D_{ckt})}{(\partial F_i / \partial D_i)} \le \Delta\gamma_{iter} \\
&\sum_{nodes\, j \in i} \frac{(\partial F_i / \partial D_i)}{(\partial F_j / \partial d_j)} \Delta\beta_j = \Delta\alpha_i \;\forall i \in P \\
&\alpha_i = \alpha_i^{init} - \Delta\alpha_i \\
&\alpha_i \ge \gamma_{target} \,,\; \Delta\alpha_i \ge 0
\end{aligned} \tag{2.18}$$

Here, we maximize the sum of the node relaxations $\Delta\beta_j$ while ensuring that the path - yield constraints are not violated. The flow for yield assignment is depicted in Figure 2.3 We start with the circuit netlist, the statistical delay library and the required yield constraint. The circuit is then sized conservatively to meet the timing yield constraint using uniform node yield assignment. A small yield decrement $\Delta\gamma_{iter}$ is chosen to ensure that the change in path criticalities is small. The node criticalities are obtained from SSTA and the LP is set up to obtain the node relaxations. The node yields are subsequently updated and the circuit is sized using the new values. This procedure is repeated until the required timing yield is obtained. The formulation in (2.15) however does not permit us to differentiate between inter and intra-chip components of variation.

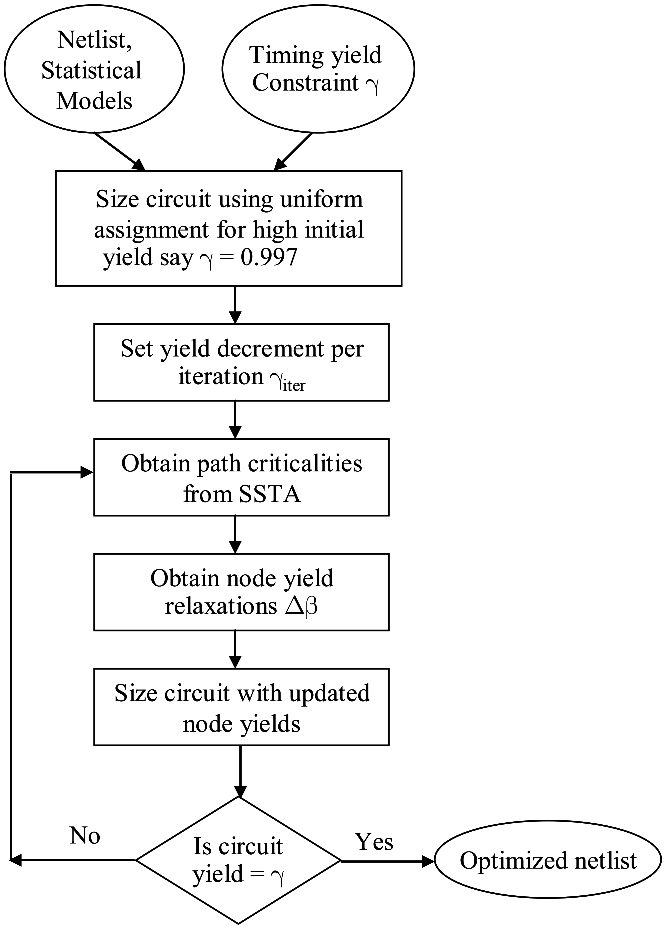

Netlist, Statistical Models
Timing yield Constraint γ
Size circuit using uniform assignment for high initial yield say γ = 0.997
Set yield decrement per iteration γ_iter
Obtain path criticalities from SSTA
Obtain node yield relaxations Δβ
Size circuit with updated node yields
Is circuit yield = γ
No
Yes
Optimized netlist

Figure 2.3: The yield assignment flow

To accomplish this, we rewrite the percent point function of gate delay $q_{d_j}(\beta_j) = \overline{d}_j + \phi^{-1}(\beta_j)\sigma_{d_j}$ such that it is linear in the inter-chip component. This transformation is equivalent to assuming that all the gates in the circuit have the same sensitivity to the inter-chip variability of the process parameters. This can be justified as inter-chip variation inherently assumes perfect correlation between devices on a chip. For the sake of exposition consider two sources of variation namely L and W . The percent point function of delay is now given by:

$$\begin{aligned}\hat{q}_{d_j}(\beta_j) = {} & \overline{d}_j + \phi^{-1}(\beta_j)((\partial d_j / \partial L)\sigma_{L_{inter}} + (\partial d_j / \partial W)\sigma_{W_{inter}}) \\ & + \phi^{-1}(\alpha)((\partial d_j / \partial L)^2\sigma^2_{L_{intra}} + (\partial d_j / \partial W)^2\sigma^2_{W_{intra}}))^{1/2}\end{aligned} \tag{2.19}$$

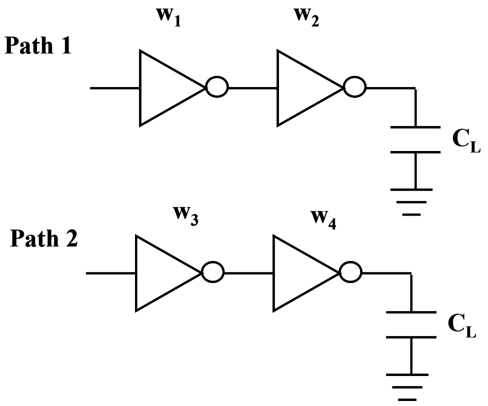


Figure 2.4: Simple circuit for evaluating yield assignment strategy

To test the strategy, we consider a simple circuit with 4 inverters and 2 paths as depicted in Figure 2.4. To better understand the effectiveness of the yield assignment strategy we consider three different scenarios. In the first scenario, when all gates are set to their minimum size, the mean delay of path 1 is smaller than that of path 2, but its variance is larger, i.e, $\mu_1 < \mu_2, \sigma_1 > \sigma_2$ Sizing the circuit for 0.997 yield results in the sizes $[s_1, s_2, s_3, s_3] = [4, 4, 3.94, 2.14]$. From Monte-carlo analysis the criticalities of paths 1 and 2 were found to be $c_1 = 0.14 \text{ and } c_2 = 0.86$. Therefore path 1 is much less critical compared to path 1. One would expect a non-uniform assignment to make take this into account and enable greater area savings on path 1. It should also ensure that the optimizer is more reluctant to downsize gates on path 2 because it is more critical. Using non-uniform assignment, the optimal decision is $[s_1, s_2, s_3, s_3] =$ [2.23, 2, 2.61, 2] and the value of the objective function is 8.87. The yield from Monte-carlo analysis is 0.95 and the criticalities of the paths are $c_1 = 0.44$ and $c_2 = 0.56$. Using uniform assignment, the value of the objective function is 8.89. The optimal decision is $[s_1, s_2, s_3, s_3] =$ [2.32, 2, 2.54, 2]. The yield from Monte-carlo analysis is 0.94 and the criticalities of the paths are $c_1 = 0.37$ and $c_2 = 0.63$. We see that our initial conjecture is indeed accurate. Non-uniform assignment is able to better optimize for yield on the critical paths and area on the non-critical paths.

In the second setting, when all gates are set to their minimum size, the mean delay of path 1 is equal to that of path 2, but its variance is larger, i.e, $\mu_1 = \mu_2, \sigma_1 > \sigma_2$. Sizing the circuit for 0.997 yield results in the sizes $[s_1, s_2, s_3, s_3] = [4, 4, 3.94, 2.5]$. From Monte-carlo analysis the criticalities of paths 1 and 2 were found to be $c_1 = 0.25$ and $c_2 = 0.75$.We observe that path1 may have been wastefully sized up in order to meet timing. After sizing using the yield assignment strategy outlined in this section the value of the objective function is 9.78 and the optimal decision is $[s_1, s_2, s_3, s_3] = [2.98, 2, 2.79, 2]$.The yield from Monte-carlo analysis is 0.94 and the criticalities of the paths are $c_1 = 0.42$ and $c_2 = 0.58$. Using uniform assignment, the value of the objective function is 9.84. The optimal decision is $[s_1, s_2, s_3, s_3] = [3.06, 2, 2.77, 2]$. Therefore non-uniform assignment is able to do better even in this scenario.

In the third setting, we want the paths to be almost equally critical after sizing, i.e, at minimum size, $\mu_1 \approx \mu_2, \sigma_1 \approx \sigma_2$ Sizing the circuit for 0.997 yield results in the sizes $[s_1, s_2, s_3, s_3] = [4, 3.94, 4, 3.38]$. As expected the criticalities are also very close to each other with $c_1 = 0.45$ and $c_1 = 0.55$. Using the non-uniform assignment strategy the value

Table 2.1: Results of non-uniform assignment on circuit with 100 inverter chains

		Area	Circuit Delay	
			μ	σ
Configuration 1	Uniform Assignment	2166	72.7	2.5
	Non-uniform Assignment	2081	72.6	2.4
Configuration 2	Uniform Assignment	1628	76.5	1.53
	Non-uniform Assignment	1608	76.5	1.52

of the objective function is 10.74. The optimal decision is $[s_1, s_2, s_3, s_3] = [3.48,\ 2,\ 3.26,\ 2]$.The yield from Monte-carlo analysis is 0.95 and the criticalities of the paths are $c_1 = 0.497$ and $c_2 = 0.503$. Using uniform assignment, the value of the objective function is 10.75. The optimal decision is $[s_1, s_2, s_3, s_3] = [3.5,\ 2,\ 3.25,\ 2]$.The yield from Monte-carlo analysis is also 0.95 and the criticalities of the paths are $c_1 = 0.48$ and $c_2 = 0.52$. Therefore, we observe that when paths have similar criticalities, both uniform and non-uniform optimizations produce almost identical results.

However, for the simple circuit we observe that the savings in area obtained by non-uniform yield assignment compared to uniform assignment is small. Our next step is to evaluate its efficacy on larger circuits. Consider a circuit comprising of 100 inverter chains. One of the paths (path 1) is more critical than the others (Configuration 1). When we compare the sizes of the gates on path 1, we see that that in the case of non-uniform assignment, the gates are sized up more, since it is more critical. The area of gates on path 1 is 15.6 as compared to 13.8 with uniform assignment. However, the off critical paths are sized down more in the case of non-uniform assignment leading to an overall

Table 2.2: Impact of logic depth, number of paths and correlation on area savings

Max. Logic Depth (m) $\rho = 0, p = 100$	6	8	10
Savings in Area (%)	3.9	3.8	3.7
Number of Paths (p) $\rho = 0, m = 6$	10	100	1000
Savings in Area (%)	3.9	3.9	3.9
$\rho = \sigma_{inter} / \sigma_{intra}$ $p = 100, m = 6$	0	1	3
Savings in Area (%)	3.9	2.1	1.2

savings in area of ~4% over uniform assignment. In the next scenario that we consider (Configuration 2), path 1 has zero criticality and the other paths are equally critical. In this case we observe that the results are nearly identical as the savings in area on the only non-critical path is not significant. Table 2.1 summarizes the results from non-uniform and uniform assignments for this circuit.

We observe that increasing the number of paths and the logic depth of the paths produces nearly identical behavior (Table 2.2). However, these experiments were performed assuming that the variability was all intra-chip. As expected when the amount of inter-chip variation is increased the effectiveness of non-uniform yield assignment decreases. Table 2.3 depicts the results obtained on the benchmark circuits. We observe that when realistic breakdowns of variability are assumed uniform assignment performs very well and the difference in the optimal solutions produced by uniform and non-uniform assignments is small. We conclude that non-uniform assignment does better than uniform assignment when paths differ in criticalities and the non-critical paths have scope for further optimization. However, when paths have similar criticalities or the non-critical paths are already at minimum size, both uniform and non-uniform optimizations produce almost identical results.

In the light of the above conclusions, we can employ a scheme of choosing the node yields uniformly in which we set all the node probabilities uniformly to $\beta_j = \beta$. Although we incur some sub-optimality, it is typically quite small, as circuit configurations produced by uniform assignment lie very close to the Pareto frontier. This value can be identified by simple line search scheme. Figure 2.5 depicts margin coefficient selection for the c880 benchmark.

Table 2.3: Results of yield assignment on benchmark circuits

	Uncorrelated			Equal Breakdown		
	A_{unif}	$A_{nonunif}$	ΔA (%)	A_{unif}	$A_{nonunif}$	ΔA (%)
C432	779	752	3.46	837	822	1.79
C888	1253	1231	1.22	1338	1328	0.74
C1908	2641	2574	2.51	2732	2696	1.30
C7552	4158	4012	3.51	4305	4236	1.58

The required circuit yield is set at 90% and the timing target is 950 ns. The minimum value of at which circuit timing becomes feasible is identified as the final node-yield assignment.

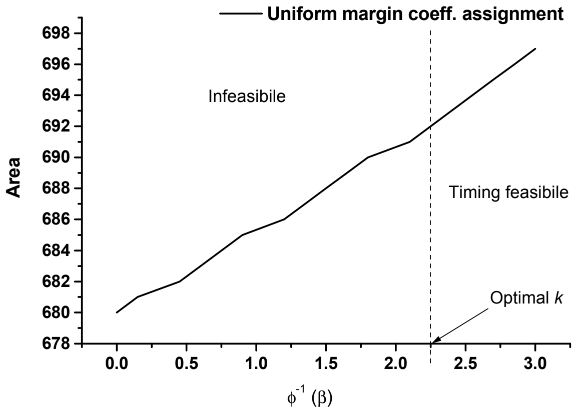


Figure 2.5: Uniform node yield selection strategy on a benchmark circuit

2.5 APPLICATION OF THE ALGORITHM IN THE MICROPROCESSOR DESIGN FLOW

Extending statistical optimization to an industrial design flow requires addressing several additional issues. The major ones are dealing with sequential elements and handling non-static logic. Even when microprocessor modules are largely implemented in static CMOS, they typically contain a number of gates implemented using non-static (pass-transistors or transmission gates) logic (i.e. multiplexers and XOR gates). These currently cannot be sized using an automated sizing algorithm. The feedback present in

```
1.Process post-synthesis structural netlist. Define:
        $G$ - Set of gates
        $F$ – Set of flops
        $G_I$ - Set of primary input gates
        $G_O$ - Set of primary output gates
2.Define $G_{FI}$ $s.t. g \in G_{FI}$ if $g \in G$ and some $input(g) \in F$ .
3.Perform breadth first search starting from each $g \in G_{FI}$ and $g \in G_I$ to
    obtain the fanout cone $FOC(g)$ terminating in $f \in F$ or $g \in G_O$ .
4. Perform backward traversal of circuit graph to levelize $F$.
   Construct the hash array $LH(f) = \{(g_i, level(g_i))\} \forall i \in output(f)$ .
5. for $j = 0$ to $max\_level$
   begin
        $NETLIST_j = \phi$
   Traverse $LH(f)$ to obtain $g_i$ $s.t. level(g_i) = j$ and obtain temp_array
   $= FOC(g_i)$ .
        Unique (temp_array)
        Augment $NETLIST_j$ with temp_array .
   end
6. Identify non-static logic gates.
7. Size $NETLIST_j$
```

Figure 2.6: The pseudo-code for application of statistical sizing to a sequential circuit.

sequential elements may render the optimization problem infeasible. Since our sizing algorithm is node based, we approach the problem of sizing a sequential circuit by extracting the combinational slices from a structural post-synthesis Verilog file. All gates between two adjacent flop-boundaries are treated as a single combinational slice. The flip-flop outputs are treated as input nodes to the combinational block the output nodes. The arrival time of signals at an input node is now given by $t_{setup} + t_{clk\text{-}Q}$ of the flop. To handle non-static logic, we adapt a simple approach. Such cells are identified and assigned the γ percentile of delay d_γ , corresponding to the size obtained from a deterministic path-based sizing heuristic, such as based on logical effort, and a realistic fanout that we take to be FO4. Here γ is the desired parametric yield. The delay d_γ can be obtained by performing a Monte Carlo simulation of the cell.

Since we fix the size of the cell, we introduce a structural constraint to capture this. We also need to make sure that the fanout gates aren't sized up such that the delay is greater than the delay d_γ. An additional constraint restricting the load on the gate is thus, also introduced. For example, if gate *j* is a multiplexer, which is typically implemented using transmission gates, the additional constraints introduced are:

$$s_j = \text{constant}, \sum s_k \leq 4s_j,\ k \in FO(j), d_j = d_\gamma \tag{2.20}$$

Here s_j is the size of the gate, $s_k, k \in FO(j)$ are its fanout gates.

The pseudo-code in Figure 2.5 describes the procedure of extracting the combinational slices from a sequential circuit. It consists of a path tracing routine to obtain the fanout cone of logic for every gate driven by a flop and a levelizing routine that assigns levels to each flop present in the circuit. We assign levels to the flops so that we can identify the gates that are present between any two flop boundaries. In this procedure, some gates may appear in different combinational slices if they appear on paths that fanout to flops at different levels. To avoid these gates being sized multiple times, the statistical sizing algorithm is applied starting with the level closest to the primary outputs. We keep track of gates that have previously been sized, and at succeeding levels, these gates are not sized again

2.6 EXPERIMENTAL RESULTS

The developed methodology was tested within an industrial microprocessor design flow for a low-power, 32-bit x86 processor, in addition to publicly available ISCAS benchmark circuits. A proprietary standard cell library targeted for a 90nm CMOS process was used. The library contains 22 cells, with more than 15 drive strengths. We used a sub-set of this library. The technology is bulk CMOS. The technology was characterized statistically with respect to three parameters: effective channel length (L_{eff}), minimum transistor width (W), threshold voltage (V_{th}). The variation was found to be 5% for L_{eff} and W, and 8% for V_{th} in terms of (σ / μ) values.

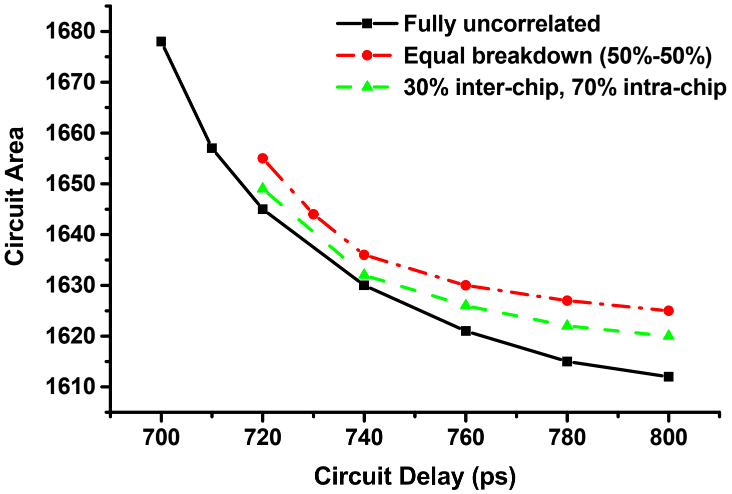


Figure 2.7: The area-delay curves for different breakdowns of variability. Statistical optimization results in smaller area for the same delay target as the intra-chip component increases

Statistical delay models were generated using a circuit simulator HSPICE by the Monte-Carlo simulation method. The SOCP algorithm was implemented using the commercially available conic solver MOSEK [57].

The area reduction that the statistical approach can enable without the loss of performance and at the same yield level is documented in Table 2.4. For the processor block, the reduction is 26%, and across the benchmark test cases the average area reduction of 19% is obtained. The analysis was performed by first optimizing the circuit using a linear optimization with uncertain parameters set to their worst case values. T_{target} corresponds to the minimum delay through the circuit obtained by unconstrained optimization in this deterministic setting and A_{det} is the corresponding area. $A_{99.99\%}$ is the area obtained by the statistical sizing algorithm for the timing constraint equal to T_{target} at the 99.99% yield level. The area savings are defined for 99.99% yield as $(A_{det} - A_{99.99\%})/A_{det}$. An equal breakdown into intra and inter-chip components is assumed for the correlated case. The application of the statistical flow to the microprocessor block was studied in depth with respect to different decompositions of process variability. Figure 2.7 shows the area-delay Pareto curves for different structures of variability. We experimented with three different breakdowns. The area improvements are better for higher ratios of intra-chip variability. As the intra-chip component increases, the sizing algorithm is able to find a configuration with lesser area for the same delay target.

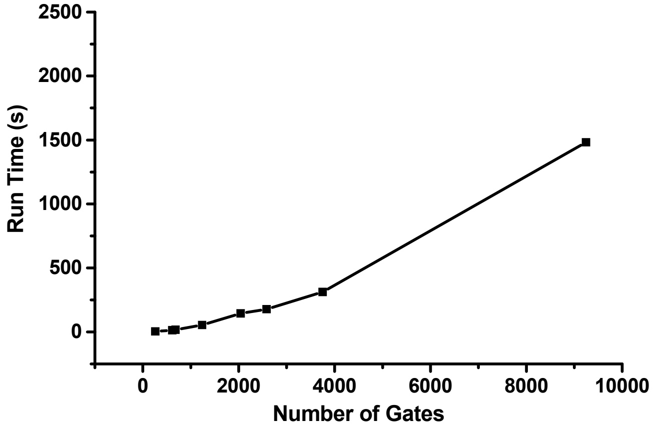


Figure 2.8: The run-time behavior of the sizing algorithm.

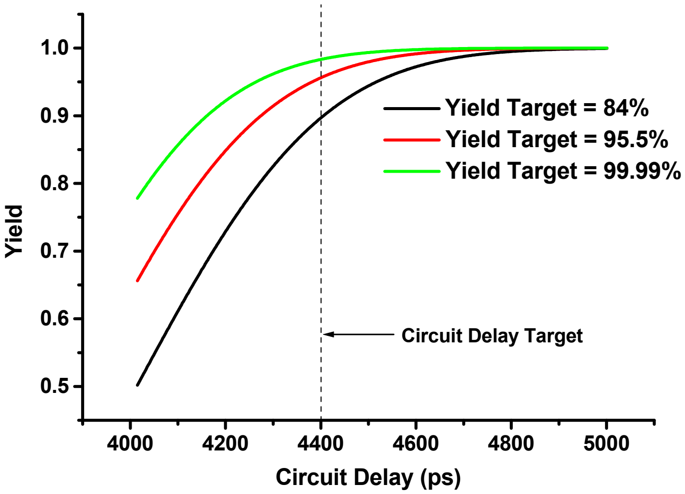


Figure 2.9: Result of performing Monte Carlo on circuit configurations for different desired yield levels for the c6288 benchmark. At target of 4.4ns, yield obtained is close to desired.

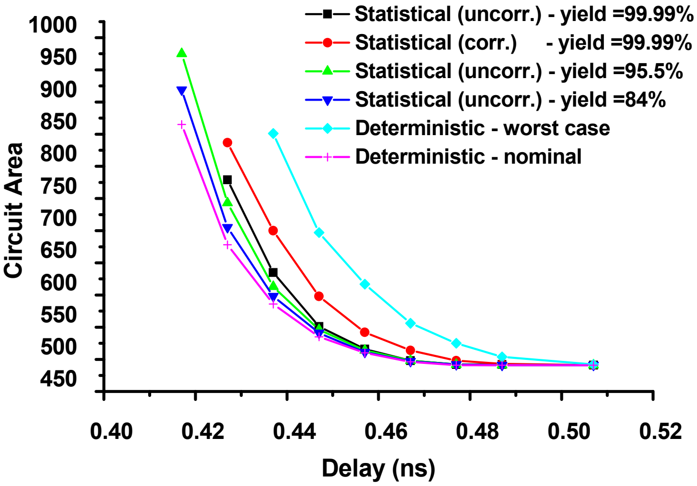


Figure 2.10: The area-delay curves at different yield levels. Statistical optimization does uniformly better than the deterministic optimization at the same yield level.

Table 2.4: Minimum area obtained by deterministic and statistical algorithms at different yield levels.

				Area - statistical optimization				Area Reduction (%)		
				Correlated	Uncorrelated					
Circuit	No. of gates	T_{target}(ns)	A_{det}	$A_{99.99\%}$	$A_{99.99\%}$	$A_{95.5\%}$	$A_{84\%}$	Uncorrelated	Correlated	Runtime (sec)
c432	261	0.427	1103	837	779	725	695	29.3	24.1	3.4
c880	615	0.475	1430	1338	1253	1244	1236	12.3	6.43	13.5
c1355	685	0.478	3828	2920	2474	2389	2327	35.3	23.7	16.4
c1908	1238	0.791	3120	2732	2641	2614	2587	15.3	12.4	53.6
c2670	2041	0.844	4166	3732	3615	3596	3578	13.2	10.4	144
c3540	2582	1.059	6406	5765	5543	5498	5457	13.4	10.0	178
c5315	3753	0.972	7988	7526	7372	7340	7309	7.7	5.7	312
c6288	2704	2.24	9647	8202	7337	7286	7198	23.9	14.9	197
μP_blk	9245	0.72	2245	1655	1642	1520	1432	26.3	26.1	1482
Average Savings								18.8	13.4	

Table 2.4 and Figure 2.8 also point to the run-time behavior of our algorithm. It can be seen that the runtime grows only slightly faster than linear in circuit size. The runtime to optimize the microprocessor block of 10K gates is reasonable, close to

24minutes. The run-time for the largest benchmark circuit is on the order of several minutes (197s).

One of the practical aspects of using the algorithm is selecting the node confidence levels to approximate the resulting circuit yield. We have found that a simple method of using the circuit confidence value directly for node values led to a surprisingly close match, as verified by a post-optimization Monte Carlo simulation on the circuit configurations obtained from the sizing algorithm (Figure 2.9 for the c6288 benchmark). The timing target was set at 4.4 ns and the circuit was sized for different yield levels. It is evident from the figure that the yield obtained is very close to what is predicted by the algorithm. The difference in area between the circuits sized at different yield levels is much greater for tighter timing constraints (~20%) and is very small at loose timing. Overall, statistical optimization performs uniformly better in comparison with the deterministic sizing. Figure 2.10 also points to the fact that in the presence of variability, certain timing targets are unachievable for a particular yield level, and designing for the nominal values of the varying parameters will lead to an unacceptably low yield. Again, this penalty grows as we approach the maximum frequency of operation of the circuit.

2.7 SUMMARY

In this chapter, we have presented a statistical sizing approach that analytically treats timing variability in timing and has good computational properties. This is due to the computationally efficient formulation of statistical sizing as a second-order conic program. We also report the first application of large-scale statistical design optimization in an industrial microprocessor design flow. In the next chapter, we examine the impact of variability on power and make a case for the consideration of both timing and power driven parametric yield loss during optimization.

Chapter 3: A Statistical Algorithm for Power- and Timing-Limited Parametric Yield Optimization

Post-synthesis circuit optimization techniques, such as sizing and dual-V_{th} allocation, are effective in reducing leakage, and have been widely explored in a deterministic setting [62][63]. While relying on different implementation strategies, all of these techniques essentially trade the slack of non-critical paths for power reduction by either downsizing the transistors or gates or setting them to a higher. In the past, case-files have been used with such optimization methods to guarantee that the circuit is optimized while guaranteeing a specific yield point. The rise of uncorrelated intra-chip variability [19][18] results in the breakdown of the case-file based approach to handling variability in optimization as it becomes impossible to come up with a case file that will guarantee a specific yield point. This requires the introduction of fully statistical optimization techniques that can handle the variance of objective and constraint functions explicitly during optimization. Given the exponential dependence of leakage power on the highly variable transistor channel length and threshold voltage, it can be expected that the introduction of rigorous statistical optimization will significantly reduce the leakage power consumption.

The primary limitation of existing statistical CAD techniques is their high computational cost. This makes the application of such algorithms to industrial-size circuits a difficult task. In this chapter, we focus on a statistical yield enhancement technique that achieves high computational efficiency, while treating both timing and power metrics probabilistically.

In order to enable an efficient computational formulation, the problem of parametric yield maximization in this algorithm is converted into that of statistical power minimization under probabilistic timing constraints. It uses a two phase approach based on optimal delay budgeting and slack utilization. The delay budgeting phase is formulated as a robust version of the power-weighted linear program that assigns slacks based on power-delay sensitivities of gates. The notion of variability in delay and power due to process variations is explicitly incorporated into the optimization, by setting up an uncertain robust linear program. The statistical (robust) linear program is cast into a second order conic program that can be solved efficiently. The slack assignment is interleaved with the configuration selection which optimally redistributes slack to the gates in the circuit to minimize total power savings.

3.1 POWER MINIMIZATION BY DELAY BUDGETING

Table 3.1 shows the tradeoffs in delay and power when transistor V_{th} is changed. Importantly, the leakage variability strongly depends on both sizing changes and V_{th} assignments. In Table 3.1 the 99% leakage corresponds to setting the threshold voltage of the device to its worst case value. A value of $\sigma / \mu = 7\%$ was used for V_{th}. It is clear that low V_{th} devices exhibit a higher leakage spread while high V_{th} devices exhibit a higher delay spread. Downsizing reduces gate area, increasing delay and reducing mean leakage, but also increasing the variance of V_{th} due to random dopant placement [64]. Thus, leakage at high quantiles can actually go up.

The general problem of gate sizing and V_{th} assignment, given that gate delay and power are non-convex, is an NP-complete problem [65], as is the extension to including multiple threshold voltages. Any computationally feasible approach to optimize circuits of any significant size will have to be based on approximating techniques. It has been shown earlier that [37] delay budgeting strategy can be used for power minimization with sizing and dual V_{th} assignment. The advantage of this formulation is that circuit modifications can be driven by global, rather than greedy, decision-making. Such a

Table 3.1: Delay and Power characteristics of low-V_{th} and high-V_{th} devices

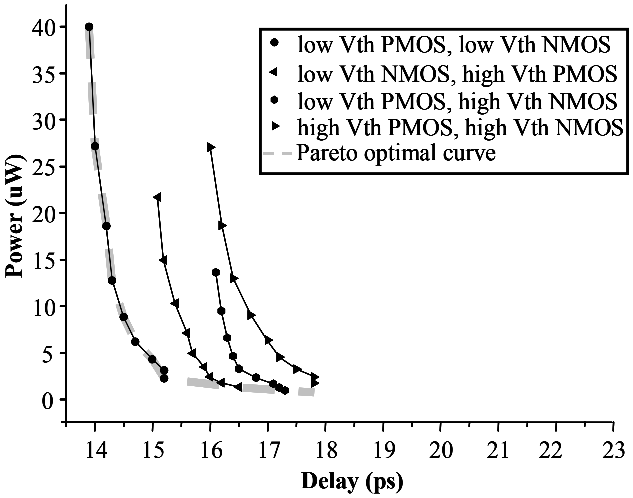
	Delay		Leakage	
	Nominal	**99th percentile**	**Nominal**	**99th percentile**
Low V_{th} (0.1 V)	1.00	1.15	1.00	2.15
High V_{th} (0.2 V)	1.20	1.50	0.12	0.20

deterministic algorithm is a two-phase iterative relaxation scheme. The input to the first phase is a circuit sized for maximum slack using a transistor (gate) sizing algorithm, such as TILOS [25], with all the devices set to low V_{th}. This circuit has the highest possible power consumption of any circuit realization. The available slack is then optimally distributed to the gates based on the power-delay sensitivities: that is, the slack is allocated in a way that maximizes the power reduction. The second phase consists of a local search among gate configurations in the library, such that slack assigned to gates in previous phase is utilized for power reduction.

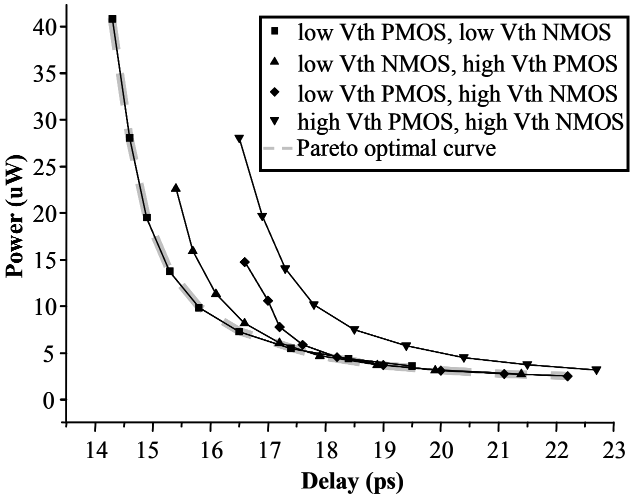
The idea of using power-delay sensitivity of a circuit as an optimization criterion is itself well known [66]. A linear measure of gate's power-delay sensitivity is power reduction per unit of added delay:

$$s = \partial P / \partial D. \tag{2.21}$$

The power reduction for an added delay $d(i)$ is then given by $s(i)d(i)$. For example, a gate driving a net with large fan-out will have a higher sensitivity than a gate with a small fan-out. Thus, a unit of added slack to a node with a higher sensitivity will lead to the greater power reduction. We rely on extending this concept to efficient optimization based on large-scale linear programming by converting a power minimization problem into a power-weighted slack redistribution problem. The notion of a gate configuration space is introduced. Let a gate configuration be any valid assignment of sizes and threshold voltages to transistors in a gate in the library. For any fixed capacitive load, a set of points in the power-delay space, i.e. Pareto points, can be identified among all the



(a) With a 5fF capacitive load and input slew 0.2ns.



(b) With a 15fF capacitive load and input slew 0.2ns.

Figure 3.1: The power-delay space for a NAND2 gate driving two different capacitive loads. The Pareto frontier is depicted by the dashed gray lines.

possible configurations (Figure 3.1). Clearly, a power optimal solution for the entire circuit will contain gates only in their Pareto-optimal configurations. Thus in optimizing the circuit we need to consider only Pareto points of all the gates. The trade-offs between delay and both leakage and dynamic power can be captured in tables, parameterized by the capacitive load. For each of the Pareto-optimal gate configurations, the decrease in power consumption (ΔP) and the change in delay (ΔD) are calculated. Thus, a sensitivity coefficient is available for every pair of Pareto points in the power-delay space. For example, we may compute the sensitivity of changing the gate from all transistors having low V_{th} to the configuration where all transistors have high V_{th}.

We may assume that initially the circuit is in its highest power-consuming state. Then, using the framework of the gate configuration space, a linear program can be formulated to distribute slack to gates with the objective of maximizing total power reduction while satisfying the delay constraints on the circuit:

$$\begin{aligned}&max \sum s_j d_j \\ &s.t.\ \ AT_j \geq AT_k + d_j^{\,0} + d_j, \text{ for } \forall k \in FI(j) \\ &AT_o \leq D_{target}, \text{ for } \forall o \in PO,\ d_j \leq \delta d.\end{aligned} \qquad (2.22)$$

Here AT_i is the arrival time at node i , T is the required arrival time at the primary output, $d_i^{\,0}$ is the delay of the gate i in the circuit configuration obtained by sizing for maximum slack, s_i is the power-delay sensitivity value for the gate, d_i is the additional slack assigned to the gate and δd is the upper bound on the slack .

The algorithm is constructed as an iterative-relaxation method. At its core is an interleaved sequence of (i) optimal slack-redistribution using LP, and (ii) the local search over the gate configuration space to identify a configuration that will absorb the assigned slack (Figure 3.2). Selection of optimal configurations is done independently for each gate. It has been shown that when the configuration space is continuous, and delay is a monotonic and separable function, such a procedure is optimal for small increments of slack assignments δd [67]. Also, the sensitivity vector is accurate within a narrow range of delay, requiring moving towards the solution under small slack increments. Even though the configuration space generated by V_{th} assignments is discrete, the ability to size transistors in a continuous manner ensures that a configuration exactly utilizing the

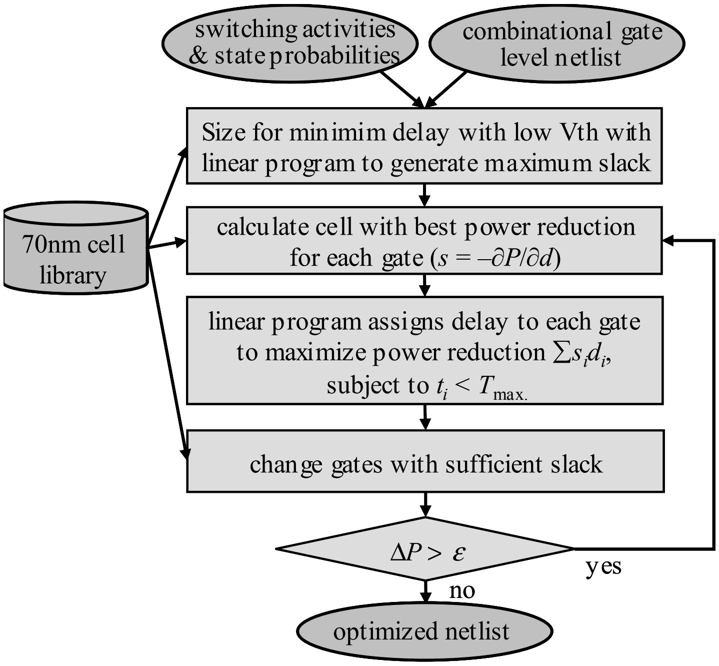

switching activities & state probabilities
combinational gate level netlist
Size for minimim delay with low Vth with linear program to generate maximum slack
70nm cell library
calculate cell with best power reduction for each gate ($s = -\partial P/\partial d$)
linear program assigns delay to each gate to maximize power reduction $\sum s_i d_i$, subject to $t_i < T_{\max}$.
change gates with sufficient slack
$\Delta P > \varepsilon$
yes
no
optimized netlist

Figure 3.2: Flowchart illustrating the algorithm for power minimization

slack allotted in the slack assignment phase can be found.

3.2 STATISTICAL MODEL FOR POWER-DELAY SENSITIVITY

The data in Table 3.1 highlights how sensitive the power and delay of individual gates to variability in process parameters. Parametric yield due to power- and timing-limited yield loss can, therefore, be substantially improved if explicit statistical treatment is extended to post-synthesis leakage power-minimization techniques based on sizing and V_{th} assignment.

We assume that both L and V_{th} follow the normal distribution, or can be easily approximated as normal. An additive statistical model that decomposes the variability, of both L and V_{th}, into the intra-chip and chip-to-chip variability components is used (Section 2.2). The relative magnitudes of the intra- and inter-chip components can be controlled by adjusting their variances. The impact of variability on delay and power is captured by statistically characterizing the power-delay sensitivity values for each gate. We now need to establish some theoretical properties of the random sensitivity vector. We assume that a first-order Taylor expansion of the gate delay function is adequate:

$$d \cong d(L_o, V_{tho}) + (\partial d / \partial L)\Delta L + (\partial d / \partial V_{th})\Delta V_{th}. \tag{2.23}$$

Note that this additive model can be extended to handle other relevant sources of variability such as gate width and T_{ox} variation.

The sub-threshold leakage current of a gate is expressed as an exponential function of the random parameters as:

$$P_{leak} = P_{leak,nom} \exp(aL + bV_{th}) \tag{2.24}$$

where $P_{leak,nom}$ is the nominal value of leakage per unit width. We obtain a good fit using this model, the *rms* error being ~8%, while the maximum error was 12%. We found that employing a model with quadratic dependence on channel length [34], L improved accuracy by < 1%, validating the use of our model. Under this model, leakage power is a log-normal random variable. Dynamic power consumption, P_{dyn} is very weakly dependent on the variation of V_{th} and L, thus it can be ignored for variational analysis.

Under this model, total power has two components, namely, P_{leak}, which is random, and the non-random component P_{dyn}. The power-delay sensitivity, thus, has to be defined separately for dynamic and leakage power.

If we define $s_d = \Delta P_{dyn} / \Delta d$ and $s = \Delta P_{leak} / \Delta d$, it is clear that s is a random variable. In our approach, we consider only this component during statistical optimization. Henceforth, the term sensitivity refers to s, with mathematical properties as described subsequently.

Theorem 3.1. Power-delay sensitivity is a log-normal random variable.

Proof: The power-delay sensitivity is defined as:

$s = \partial P_{leak} / \partial d$.

Here, the implicit assumption is that the variability in delay arises due to variation in V_{th} and L.

$$\begin{aligned} s &= P_{leak,nom} e^{aL+bV_{th}} \frac{\partial}{\partial d}(aL + bV_{th}) \\ &= P_{leak,nom} e^{aL+bV_{th}} (\frac{a}{(\partial d / \partial L)} + \frac{b}{(\partial d / \partial V_{th})}) \\ &= (\frac{a}{(\partial d / \partial L)} + \frac{b}{(\partial d / \partial V_{th})}) P_{leak}. \end{aligned} \tag{2.25}$$

Since the sensitivity terms $\partial d / \partial L$ and $\partial d / \partial V_{th}$ are not random variables, being independent of V_{th} and L, the sensitivity s follows a log-normal distribution. □

Based on the above variability models, the variance and covariance of the power-delay sensitivity coefficients are characterized via a Monte-Carlo simulation for all the cells in the library. For each cell, delay and power are represented statistically using (2.23)and (2.24) . The statistical properties of the power-delay sensitivity of the cell can then be computed using (2.25).

Setting $c = \frac{a}{(\partial d / \partial L)} + \frac{b}{(\partial d / \partial V_{th})}$, the variance of power-delay sensitivity for a cell can be expressed as:

$$\begin{aligned} \mathrm{var}[s] &= \mathrm{var}[(\frac{a}{(\partial d / \partial L)} + \frac{b}{(\partial d / \partial V_{th})}) P_{leak}] \\ &= c^2 \, \mathrm{var}[P_{leak}]. \end{aligned} \tag{2.26}$$

Similarly the covariance between two cells can be obtained as:

$$\mathrm{cov}[s_i, s_j] = c_i c_j \, \mathrm{cov}[P_{leak,i}, P_{leak,j}]. \tag{2.27}$$

The characterization thus provides the numerical values of the vector of mean sensitivities, $\overline{s}$ and the covariance matrix Σ of s. To simplify modeling, we assume that this correlation arises due to die-to-die variation, and not from spatial correlation between cells on a die.

In the presence of non-zero inter-chip variability and spatial intra-chip variability, the sensitivity coefficients are correlated. Because the optimization is easier to set up when the sensitivities are uncorrelated, Principal Component Analysis (PCA) is used to transform the original vector of sensitivities into one with a diagonal covariance matrix. This transformation handles cell correlation arising from V_{th} and L. Given the covariance matrix Σ of the vector of sensitivities s, PCA obtains the vector of principal components s'. Then, the sensitivities are expressed in terms of their uncorrelated principal components [68]:

$$s = \overline{s} + As'. \tag{2.28}$$

where $\overline{s}$ is the vector of mean sensitivities and the matrix A is the eigenvector matrix of Σ.

3.3 DELAY BUDGETING USING ROBUST LINEAR PROGRAMMING

In this section, a statistical equivalent for the power minimization strategy is described. To handle variability of process parameters, the problem is reformulated as a robust linear program. As mentioned in Section 3.2, the algorithm assumes two primary sources of variability: effective channel length (L) and gate-length independent variation of threshold voltage (V_{th}). This modeling framework gives the ability to account for the contrasting effects of parameter variability on low- and high- gates: low-V_{th} gates exhibit higher variation in leakage, while high-V_{th} gates exhibit higher delay variability.

When formulating a statistical power minimization problem, we find that an equivalent formulation of (2.22), which places the power weighted slack vector into the constraint set, is more convenient. Here, we define equivalence in the following manner: given the same power and timing constraints, the optimal solution, i.e, the vector of slacks produced by the two LPs is the same. In resorting to this definition, we restrict ourselves to the intersection of feasibility sets of the two LPs.

Theorem 3.2. If $\overline{P}$ is the initial power consumed by the circuit, $\hat{P}$ is the optimal power achieved by (2.22) at a specific D_{target} and $\hat{d}_1$ the corresponding vector of optimal allocated slacks, the optimization problem (2.29) is equivalent to (2.22).

$$\begin{aligned}&min \sum d_j\\&s.t. \ \sum s_j d_j \geq \overline{P} - \hat{P}\\&AT_o \leq D_{target}, \ \text{ for } \forall o \in PO\\&AT_j \geq AT_k + d_j^{\,0} + d_j, \ \text{ for } \forall k \in FI(j), d_j \leq \delta d.\end{aligned} \tag{2.29}$$

Proof: We need to show that, if $\hat{\boldsymbol{d}}_2$ denotes the optimal vector of allocated slacks from (), and $P(\hat{\boldsymbol{d}}_2)$ is the corresponding minimum power solution at the specified $T_{\max}$, $P(\hat{\boldsymbol{d}}_1) = \hat{P} = P(\hat{\boldsymbol{d}}_2)$. Since we start with the same initial configuration $\overline{P}$, it suffices to prove that $\hat{\boldsymbol{d}}_1 = \hat{\boldsymbol{d}}_2$.

Representing $\sum s_i d_i$ by $\boldsymbol{s}^T \boldsymbol{d}$, where $\boldsymbol{s}$ is the vector of sensitivities and $\boldsymbol{d}$ is the vector of assigned slacks. It follows that,

$$\boldsymbol{s}^T \hat{\boldsymbol{d}}_1 \geq \boldsymbol{s}^T \hat{\boldsymbol{d}}_2. \tag{2.30}$$

since $\hat{\boldsymbol{d}}_2$ is a sub-optimal solution to (2.22). Let $\overline{P} - \hat{P} = \Delta P$. Feasibility of (2.29) implies:

$$\begin{aligned}&\boldsymbol{s}^T \hat{\boldsymbol{d}}_2 \geq \overline{P} - \hat{P}\\&\Rightarrow \Delta P \leq \boldsymbol{s}^T \hat{\boldsymbol{d}}_2.\end{aligned} \tag{2.31}$$

But from our assumption $\Delta P = \boldsymbol{s}^T \hat{\boldsymbol{d}}_1$ since this is the optimal power savings enabled by (2.22). Therefore, from (2.31):

$$\boldsymbol{s}^T \hat{\boldsymbol{d}}_1 \leq \boldsymbol{s}^T \hat{\boldsymbol{d}}_2. \tag{2.32}$$

Equations (2.30) and (2.32)together must imply:

$$\boldsymbol{s}^T \hat{\boldsymbol{d}}_1 = \boldsymbol{s}^T \hat{\boldsymbol{d}}_2. \tag{2.33}$$

And, $s_i \neq 0 \, \forall i$ implies $\hat{\boldsymbol{d}}_1 = \hat{\boldsymbol{d}}_2$. □

From a physical perspective, the minimization in the objective function of (2.29) forces the LP to place a premium on the total slack and assign more slack to gates with higher sensitivity in order to meet the power constraint.

The statistical equivalent of (2.29) is now formulated by probabilistically treating the uncertainty of the sensitivity vector and of timing constraints:

$$\begin{aligned}
&min \sum d_j \\
&s.t.\ \ P\left(\sum s_j d_j \geq P_{max} - P_{const}\}\right) \geq \eta \\
&P(AT_o \leq D_{target}) \geq \gamma \text{ for } \forall o \in PO \\
&AT_j \geq AT_k + d_j^{\,0} + d_j,\ \text{ for } \forall k \in FI(j).
\end{aligned} \tag{2.34}$$

In this formulation, the change in dynamic power, summed across all the cells, due to additional slack assigned to each cell, is a deterministic quantity, it can be subtracted out at each iteration.

3.4 STATISTICAL DELAY BUDGETING USING SOCP

In (2.34), the deterministic constraints have been transformed into the probabilistic constraints. These probabilistic constraints set respectively the power-limited parametric yield, η, and the timing-limited parametric yield, γ . Based on the formulation of the model of uncertainty, they capture the uncertainty due to process parameters via the uncertainty of power and delay metrics. We now transform both probabilistic inequalities such that they can be efficiently handled by the available optimization methods. The challenge is to handle these inequalities analytically, in closed form.

3.3.1 Transforming the Circuit Timing Constraint

Using the theory proposed in Section 2.4, the node yields can be obtained with aid of SSTA. The probabilistic timing constraints can be written as:

$$\begin{aligned} &AT_o \leq D_{target}, \quad \text{for } \forall o \in PO \\ &AT_j \geq AT_k + d_j^{\,0} + k_j \sigma_{d_j^0} + d_j, \ \text{for } \forall k \in FI(j). \end{aligned} \tag{2.35}$$

Where $k_j = \phi^{-1}(\beta_j)$, β_j is the node yield assigned.

3.3.2 Handling the Probabilistic Power Constraint

Letting $u = \sum s_i d_i = s^T d$, $\Delta P = P_{max} - P_{const}$ and $\eta' = 1 - \eta$, we can re-write the probabilistic constraint as $P(\ln u \leq \ln \Delta P) \leq \eta'$. In Section 3.2 we have shown that *u* can be modeled as a lognormal random variable. If $u \sim LN(m, \delta^2)$, then, $\ln u \sim N(\mu, \sigma^2)$. Now, if the mean of *u* is *m* and the standard deviation of *u* is δ, then, $\mu = \ln\left(m^2 / \sqrt{m^2 + \delta^2}\right)$, $\sigma = \sqrt{\ln(1 + m^2 / \delta^2)}$.

The translation-invariance property of a normal distribution can be used to express as:

$$P\left(\frac{\ln u - \mu}{\sigma} \leq \frac{\ln \Delta P - \mu}{\sigma}\right) \leq \eta'. \tag{2.36}$$

Since $(\ln u - \mu)/\sigma \sim N(0,1)$, letting $\phi(.)$ be the *cdf* of $N(0,1)$, the constraint $P(\ln u \leq \ln \Delta P) \geq \eta'$ can be expressed as:

$$\mu + \phi^{-1}(\eta')\sigma \geq \ln(\Delta P). \tag{2.37}$$

Using the above relationships between *m* and *μ*, and *σ* and *s*, we can express the probabilistic constraints as:

$$\ln\left(m^2 / \sqrt{m^2 + \delta^2}\right) + \phi^{-1}(\eta')\sqrt{\ln(1 + m^2 / \delta^2)} \geq \ln(\Delta P) \tag{2.38}$$

The advantage of this our formulation is the ability to take into account uncertainty of the constraint function explicitly. Indeed, the mean of *u* is $m = E(s^T d) = \overline{s}^T d$, and the variance is $\delta^2 = d^T \Sigma d$, where Σ is the covariance matrix of the vector of sensitivities *s*. Using the above non-linear probabilistic constraint, however, would require solving a non-linear optimization problem which is computationally expensive. However, we can reformulate this problem as a second-order conic program (SOCP) that can be solved efficiently.

Letting $\theta = \phi^{-1}(\eta')$ and using () we can define:

$$f_0(m,\delta,\theta) = \ln(m^2 / \sqrt{m^2 + \delta^2}) + \theta\sqrt{\ln(1 + m^2 / \delta^2)}. \tag{2.39}$$

To formulate (2.39) as an SOCP, we need a percent point function which is linear in m and δ. To this end, a least square of fit of f_0 onto f, which is linear in these parameters, has to be performed.

To perform this fit, we make use of an interesting property of the lognormal distribution, namely, its shape parameter. The shape parameter is the standard deviation σ of the underlying normal random variable. This dictates the broadness of the lognormal distribution [69] . In practice, we observed that the leakage and sensitivity distributions had shape parameters $\sigma < 0.5$. Therefore, we are justified in confining the region of the approximation to $\sigma \leq 0.5$. Normalizing the mean μ, we sample the shape parameter σ and obtain the corresponding values of m and δ using the relations:

$$\begin{aligned} m &= \exp((2\mu + \sigma^2)/2) \\ \delta &= \sqrt{\exp(2\mu + 2\sigma^2) - \exp(2\mu + \sigma^2)}. \end{aligned} \tag{2.40}$$

From (2.40), a set of values can be computed for f_0 corresponding to the different shape parameters. These can then be used to fit the linear function f.

$$f = (m + \kappa(\theta)\delta)\lambda(\theta). \tag{2.42}$$

Here, $\lambda(\theta)$ and $\kappa(\theta)$ are linear functions of θ. The *rms* error was found to be ~5% and the maximum error was ~9%.

Using $m = \overline{s}^T d$ and $\delta^2 = d^T \Sigma d$, where Σ is the covariance matrix of the vector of sensitivities *s,* the constraint can now be re-written as:

$$\overline{s}^T d + \kappa(\theta)(d^T \Sigma_s d)^{1/2} \geq \ln(\Delta P) / \lambda(\theta). \tag{2.43}$$

Using (2.43), and (2.35)we can formulate the SOCP as:

$$\begin{aligned}&min\sum d_j\\&s.t.\ \overline{s}^T d+\kappa(\theta)(d^T\Sigma_s d)^{1/2}\geq \ln(\Delta P)/\lambda(\theta)\\&AT_j\geq AT_k+d_j^{\,0}+k_j\sigma_{d_j^{\,0}}+d_j,\ AT_o\leq D_{target}.\end{aligned}\tag{2.44}$$

Here $\theta=\phi^{-1}(\eta')$, and the node margin coefficients k_i are obtained using the yield assignment strategy outlined in Section 2.4.

3.5. EXPERIMENTAL RESULTS

The algorithm was implemented in C as a pre-processing module to interface with a commercial conic solver available as part of MOSEK [57]. The benchmark circuits were synthesized to a cell library that was characterized for a 70nm process using Berkeley Predictive Technology Model [61]. The algorithm was run on a dual core 1.5GHz. AMD Athlon machine with 2GB of RAM.

The gates present in the library are NOR2, NOR3, NOR4, NAND2, NAND3, NAND4 and inverter. Gates have eight discrete sizes, ranging from 1× to 8× the minimum size, and were characterized for a fixed input slew of 20ps, based on output slew observed for an FO4 inverter, characterized using SPICE. Though in its current form our approach cannot capture the impact of slew on delay at, it is possible to model the dependence of gate delay (and output slew) on input slew linearly [70]. Such a model can be easily accommodated in our framework. Gate delay (average of worst case rise and fall delay) and internal power were specified by lookup tables versus load capacitance. No wire loads were used, but it would be straightforward to add these to the load capacitance. Switching power was calculated as normal ($\alpha f C_L V_{dd}^2$, where α is the activity factor, f is the clock frequency, C_L is the load capacitance, and V_{dd} is the supply voltage). Leakage power is specified by gate input state as:

$$P=(1-\alpha)\sum_i P_i\beta_i.\tag{2.45}$$

where P_i is the leakage current for a gate in dominant leakage state *i*, and β_i is the probability that the gate is in that dominant state. The activity factors and state probabilities were determined by random simulation. It is assumed that granularity of V_{th} allocation is at the NMOS/PMOS stack level. For NMOS (PMOS) transistors, the high threshold voltage is 0.20V (–0.20V) and the low threshold voltage is 0.10V (–0.10V).

Different levels of variability in L were explored ranging from 3% to 8% of σ/μ. It is assumed that $\sigma_{V_{th}}$ of a gate is inversely proportional to its size, and gate-length independent V_{th} variation is due to random dopant placement. Pelgrom's model [64] is used to describe $\sigma_{V_{th}}$ dependence on transistor size. The assumed magnitude of V_{th} variability is σ / μ = 7%. The mean and covariance matrix of cell sensitivities were computed for all gate configurations using SPICE. Different structures of variability of the process parameters were explored. In one scenario, we considered all variation to be uncorrelated, in the second case we assumed an equal breakdown of total variability into its inter and intra-chip components. In another experiment, inter-chip variation was assumed to be the dominant component. Principal component analysis was used to

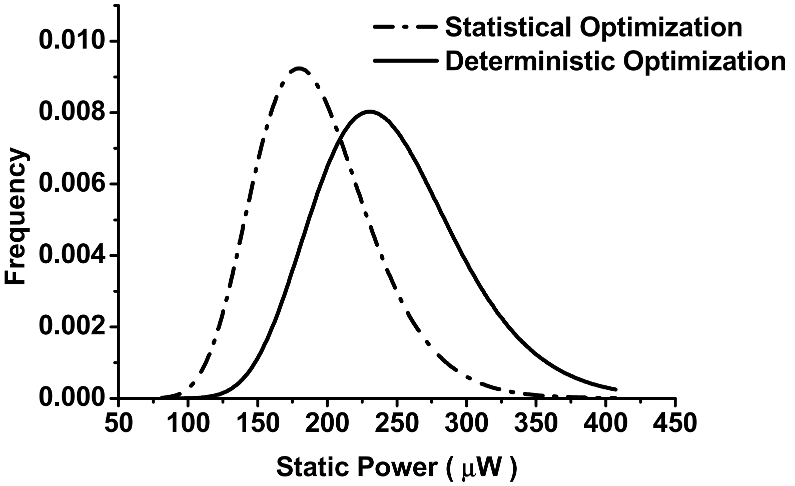


Figure 3.3: PDFs of static (leakage) power produced by a Monte- Carlo simulation of the benchmark circuit (C432) optimized by the deterministic and statistical algorithms.

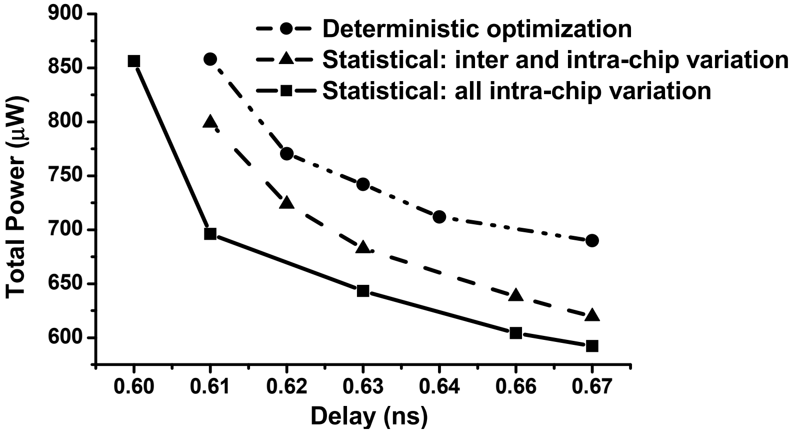


Figure 3.4: Power-delay curves for 99.9% timing and power yield.

orthogonalize the covariance matrix of cell sensitivity coefficients. The performance and run-time behavior of the optimization algorithm is validated on the public ISCAS'85 benchmark circuits and several industrial blocks. All comparisons are done for the same arrival time at the primary output. This can be achieved by performing the deterministic power optimization under identical statistical timing constraints. Deterministic optimization in this case refers to optimization where the random parameters L and V_{th} are set to the worst case values.

3.4.1 Effectiveness of Algorithm in Optimizing Power

Across the benchmarks results indicate that the savings of, on average, 33% in leakage power (measured at the 99.9th quantile) without the loss of timing or power yield can achieved by statistical optimization as opposed to the deterministic approach, (Table 3.2). The level of L_{eff} variability is assumed to be σ / μ = 8%. In the table, n is the number of gates in the circuit, and Static and Total refer to static and total power in μW respectively. The results in Table 3.2 are for the case where variability is evenly decomposed into its intra and inter-chip components. Table 3.2 also documents the run-

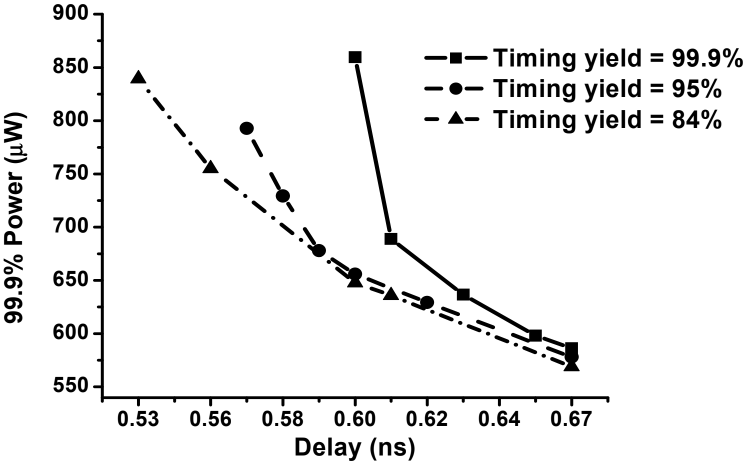


Figure 3.5: Power-delay curves at different timing yield levels for the C432 benchmark.

time behavior of the statistical optimization algorithm. For the largest benchmark the run-time is of the order of a few (~4) minutes. It is pertinent to mention that the speedup is obtained due to the special structure of the SOCP program that is not available to the general non-linear solvers enabling the optimization problem to be solved extremely efficiently.

The reason for the reduction in power enabled by statistical optimization is the ability of the statistical algorithm to explicitly account for the variance of constraint and objective functions. This can be attributed to the fact that the statistical optimization allots slack more efficiently. One manifestation of the superiority of statistical optimization is the fact that it can assign more transistors to a high V_{th}. For example for the C432 benchmark optimized for a target delay of 0.55ns for 99.9% timing and power yields, the number of transistors set to high V_{th} by the statistical algorithm is 20% more than the corresponding number for the deterministic algorithm. As a result, the spread of the leakage distribution is reduced and the mean is shifted towards lower values. Figure 3.3 shows the *pdf* of static power obtained by a Monte Carlo simulation of the circuit configurations produced by the statistical and deterministic optimizations. Both the mean

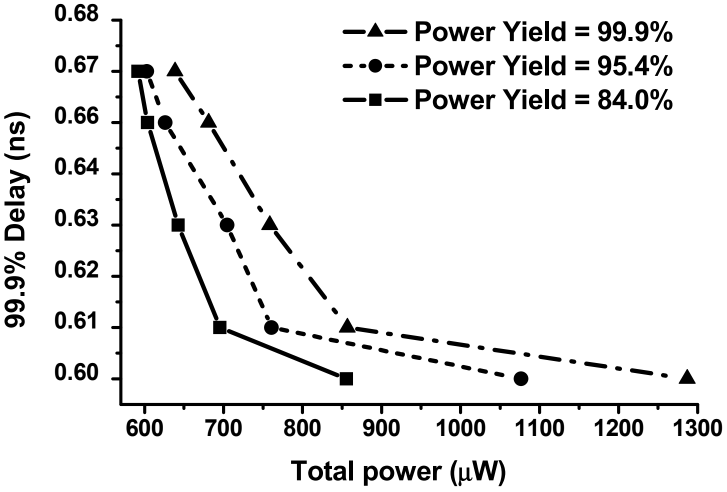


Figure 3.6: Power-delay curves at different power limited yields and variance of static power for the deterministically optimized circuit are greater, which implies that the static power savings increase at higher percentiles.

The superiority of statistical optimization over the deterministic optimization is illustrated in Figure 3.4. Under the same power and timing yield constraints ($\gamma = \eta =$ 99.9%), statistical optimization produces uniformly better power-delay curves. The improvement strongly depends on the underlying structure of physical process variation. As the amount of uncorrelated variability increases, i.e. the intra-chip component grows in comparison with the chip-to-chip component, the power savings enabled by statistical optimization increase. The power savings at the 95th percentile are 23%, and those at 99th percentile are 27% respectively.

The ability to directly control the level of parametric power and timing limited yield permits choosing a 'sweet spot' in the power-delay space. Figures. 3.5-3.6 show aset of power-delay curves for one of the benchmarks, c432. Figure 3.5 plots the total power vs. delay at the output obtained by running the statistical optimization for various timing yield levels (γ), with the power yield set at 99.9%. It can be observed that at tight timing constraints the difference in power optimized for different yield levels is significant. Figure 3.6 confirms that optimizing the circuit for a lower power yield will lead to higher total power consumption and longer delay. For the same yield, the trade-off between power and arrival time is much more marked at tighter timing constraints.

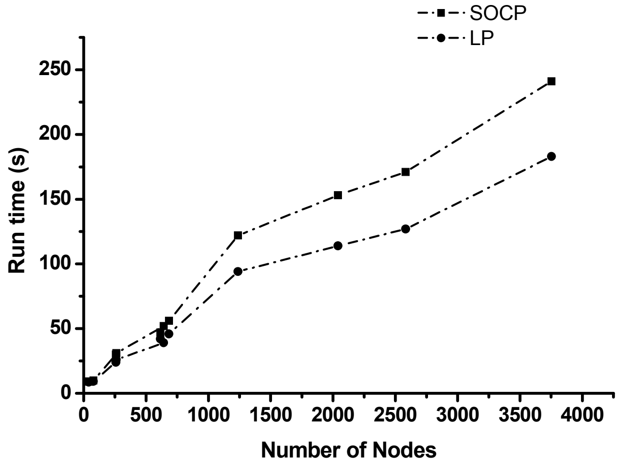


Figure 3.7: Run time behavior of the statistical power optimization algorithm

Table 3.2: Power Savings Obtained by Statistical over Deterministic Algorithm

		Timing yield $\zeta = 99.9\%$, Power yield $\eta = 99.9\%$						Timing yield $\zeta = 84\%$, Power yield $\eta = 99.9\%$						
	n	Deterministic Optimization		Statistical Optimization		Savings in Power (%)		Deterministic Optimization		Statistical Optimization		Savings in Power (%)		Run Time (s)
		Static	Total	Static	Total	Static	Total	Static	Total	Static	Total	Static	Total	
sc_ivlogic	40	29	140	19	111	35.2	20.8	19	113	12	97	33.3	14.8	9
sc_inc12	78	45	218	28	176	37.7	19.4	32	192	21	149	35.0	22.0	10
sc_edcs1	258	186	747	127	632	32.1	15.4	126	683	87	583	30.7	14.6	30
c432	261	157	858	107	696	32.3	18.9	112	783	75	620	32.8	20.8	31
c499	641	457	1290	305	1066	33.4	17.3	302	1054	213	894	29.6	15.2	52
c880	615	713	1217	492	1018	31.0	16.3	461	847	331	728	28.2	14.1	47
c1355	685	531	1501	343	1216	35.5	19.0	379	1240	244	994	35.6	19.8	56
c1908	1238	899	2559	611	2112	32.1	17.5	673	2284	503	1945	25.2	14.9	122
c2670	2041	1468	4814	1055	4113	28.1	14.6	1112	3926	813	3382	26.9	13.9	153
c3540	2582	1181	5549	809	4765	31.5	14.1	856	4498	602	3943	29.7	12.3	171
c5315	3753	2984	5411	1960	4493	34.3	17.0	2096	3769	1456	3222	30.5	14.5	241
c6288	2704	1178	5744	778	4691	34.0	18.3	746	4130	529	3429	29.1	17.0	273
Average savings						33.1	17.4					30.5	16.2	

3.4.2 Computational Properties of the Algorithm

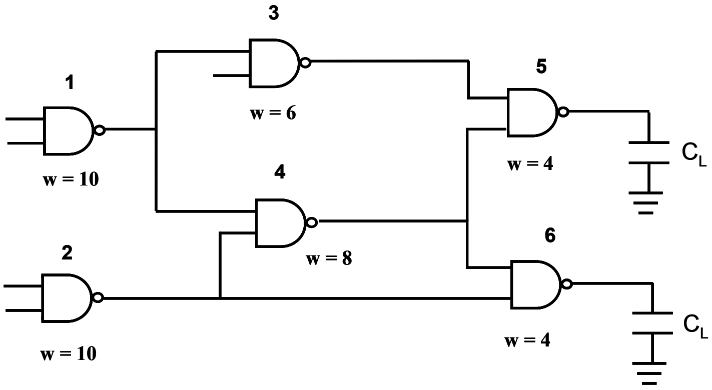
Figure 3.7 depicts the run-time behavior of the algorithm. The optimization problems were solved using the interior point optimization package MOSEK. A single SOCP optimization run of c6288 for slack assignment takes about 11 seconds. It can be seen that the run-time is roughly linear in circuit size making the algorithm scalable to large industrial blocks. The complexity of the second phase of the power minimization algorithm, which maps the allotted slack to gates in the library is $O(mN)$, where m is the number of alternatives in the gate configuration space. The overall complexity of our statistical power minimization algorithm is, therefore, close to linear

As mentioned previously, the granularity of V_{th} allocation to gates in the library is at the NMOS/PMOS stack level. Since each gate type has 8 posible sizes, a gate in the library has 32 possible configurations. We ran the algorithm using a smaller library, by having only four possible gate sizes and restricting the V_{th} allocation to gate level (i.e. all transistors in a gate have the same V_{th}) and found that this led to a solution which consumed more power for the same delay constraint. However, the benefits with regards to run time were minimal, as the mapping phase does not limit the run-time performance of the algorithm.

3.5 A CASE STUDY ON THE C17 BENCHMARK

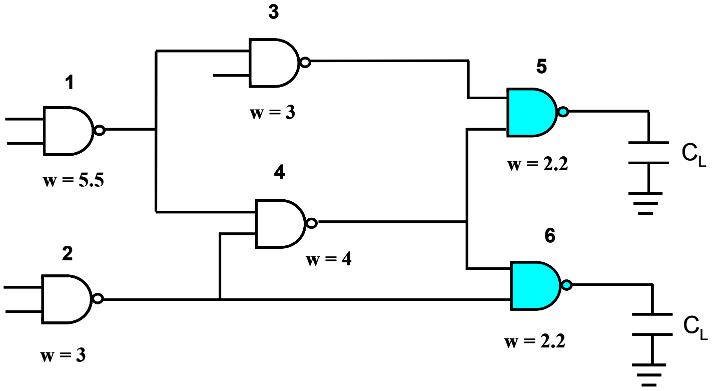
In this section we illustrate how statistical optimization is able to achieve savings in dynamic and leakage power compared to deterministic optimization with the help of the simple c17 benchmark circuit. The circuit configuration which results in minimum delay (and maximum slack) is depicted in Figure 3.8. All the gates are at low-V_t.

The circuit configuration produced by deterministic optimization is shown in Figure 3.9. We note the gates 5 and 6 have been set to high-V_t.



Circuit delay at 99.97 quantile ($D_{99.97}$) : 42.5
Dynamic power (P_{dyn}) : 26.1
Leakage power at 99.97 quantile($P_{leak,99.97}$) : 12.5
Low-Vt device width : 42
High- Vt device width : 0

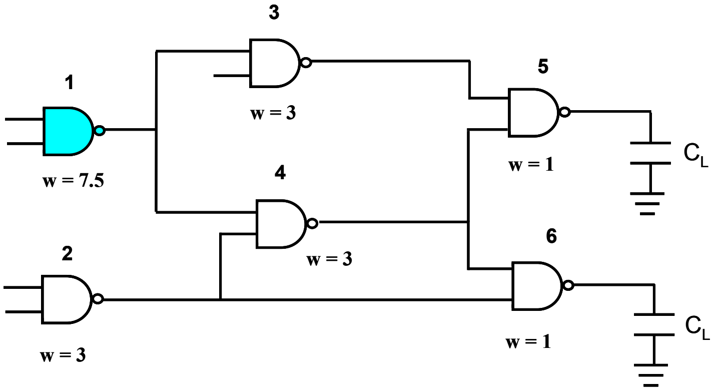
Figure 3.8: c17 circuit sized for maximum slack



Circuit delay at 99.97 quantile ($D_{99.97}$) : 47.5
Dynamic power (P_{dyn}) : 12.4
Leakage power at 99.97 quantile ($P_{leak,99.97}$) : 6.4
Low-Vt device width : 15.5
High- Vt device width : 4.4

Figure 3.9: c17 circuit configuration produced by deterministic optimization

The following configuration is produced by statistical optimization. We note the gates 5 and 6 are retained at low- V_t, but gate 1 is set to high- V_t.



Circuit delay at 99.97 quantile ($D_{99.97}$) : 47.5
Dynamic power (P_{dyn}) : 11.1
Leakage power at 99.97 quantile ($P_{leak,99.97}$) : 5.48
Low-Vt device width : 11
High- Vt device width : 7.5

Figure 3.10: c17 circuit configuration produced by statistical optimization

The differences in statistical and deterministic optimization can be explained in the following way. For large gates, the statistically feasible alternative is a gate with higher threshold voltage, and slightly higher drive strength. This is because going from a higher threshold voltage to a lower threshold voltage results in a significant reduction in the variance of leakage. Statistical optimization assigns slacks in a way that picks configurations with maximum mean sensitivity but minimum variance of sensitivity. In the deterministic case however, slack is assigned in a way that results in the gate with the maximum mean value of sensitivity being picked. This is why gate 6 is set to a higher size but also higher threshold voltage in the case of statistical optimization

Now consider gates 5 and 6. We see that statistical optimization picks a smaller gate but a with lower threshold voltage while deterministic optimization picks a high-V_t

gate with larger drive strength. The leakage penalty for choosing a low-V_t gate in this case is very small, and is compensated by the smaller value of dynamic power. However, low-V_t gates have smaller variance in delay than high-V_t gates. This means that the statistically feasible alternative in such cases is a low-V_t gate with smaller drive strength. One can therefore think of statistical optimization as leveraging its ability to account for variance in delay to save dynamic power. A manifestation of these two mechanisms is exemplified by a larger high-V_t device width and smaller low-V_t device width, resulting in net savings in leakage power and smaller total device width leading to savings in dynamic power for statistical optimization compared to deterministic optimization

3.6 SUMMARY

In the recent past it was sufficient to model the impact of variability on timing. With high-end designs experiencing a double-sided squeeze on parametric yield due to the power-dissipation limits, power variability needs to be explicitly taken into account. This requires the adoption of new analysis and optimization methodologies that incorporate the notion of power-limited parametric yield loss. In this chapter we have presented a novel statistical algorithm for total power minimization that is based statistical slack budgeting using second order conic programming. The algorithm is capable of treating both power and timing metrics probabilistically, allowing joint optimization of both power and timing limited yield. The algorithm can handle multiple sources and different structures of variability. We demonstrate that across the benchmarks the algorithm achieves significant reduction in static and total power. In the next two chapters, we present joint design-time and post-silicon techniques that address the primary limitation of design time techniques: their inability to react to conditions on chip after manufacture.

Chapter 4: Joint Design-Time and Post-Silicon Minimization of Parametric Yield Loss using Adjustable Robust Optimization

The fundamental limitation of design-time methods is that they impose an overhead on each instance of the fabricated chip since they intrinsically lack the ability to "react" to the actual conditions on the chip. For example, when using sizing for timing optimization they impose a fixed area overhead that may be wasteful on some instances of the ICs that would meet timing even with smaller driver sizes. Having an adjustable-width driver would be ideal, since it could ensure meeting constraints with the minimum overhead for each chip.

The problem that we address in this chapter is how to perform design-time circuit optimization and post-silicon tuning *jointly*. Why should these two steps be coordinated, i.e., why do we need joint co-optimization? The two methods operate from different viewpoints: in design-time optimization a decision (e.g., sizing) must be made *before* the realization of uncertainty (gate length), while in post-silicon tuning of the decision (the value of bias to apply) is made *after* the realization of uncertainty, i.e., when the chip's physical properties have been determined during manufacturing.

However, the two paradigms operate within a single budget of uncertainty, and thus meeting constraints can be achieved by both methods. But their cost-effectiveness depends on specific conditions, such as the spatial correlation of process variability, the granularity of adaptivity that can be implemented, and the magnitude of leakage power in comparison with the switching power. The objective of this chapter is to develop formal means and optimization methods that will allow joint optimization. The specific optimization strategy will jointly consider the amount of variability and cost-effectiveness of power reduction strategies, to derive *a policy* that will guide post-silicon tuning, as well as make the first-phase design decisions. This will allow to optimally partition the design space between these levels of hierarchy.

Formally, the objective of the algorithm we develop is to minimize the expected value of leakage power under a given delay constraint T at a given yield α:

$$\min\ E_{leak}\ s.t. P(D \le T) \ge \alpha \tag{4.1}$$

This formulation is generic and, different specific optimization mechanisms can be studied. In this chapter we focus on sizing and adaptive body bias for threshold control at the chip level, with only a small number of partitions of the chip into individually tunable clusters. . In the above formulation, the objective function and the constraints depend on both the design time optimization variables (sizes) and the post silicon decision variables (body biases). The problem can be formally viewed as a two-phase optimization under uncertainty with recourse. The key contribution of our approach is the derivation of the optimal policy for body biasing as an affine function of the realizations of the uncertain parameters (gate length L and threshold voltage V_{th}). The solution to the above optimization problem therefore yields the sizes for the gates in the circuit and an optimal body bias policy.

4.1 GATE AND CIRCUIT MODELING

4.1.1 Delay and Leakage Models

Adjusting the circuit properties to manufacturing conditions can be achieved by several techniques, including adaptive buffer sizing, adaptive body biasing, and adaptive supply voltage biasing. Because the joint timing-leakage optimization is of primary

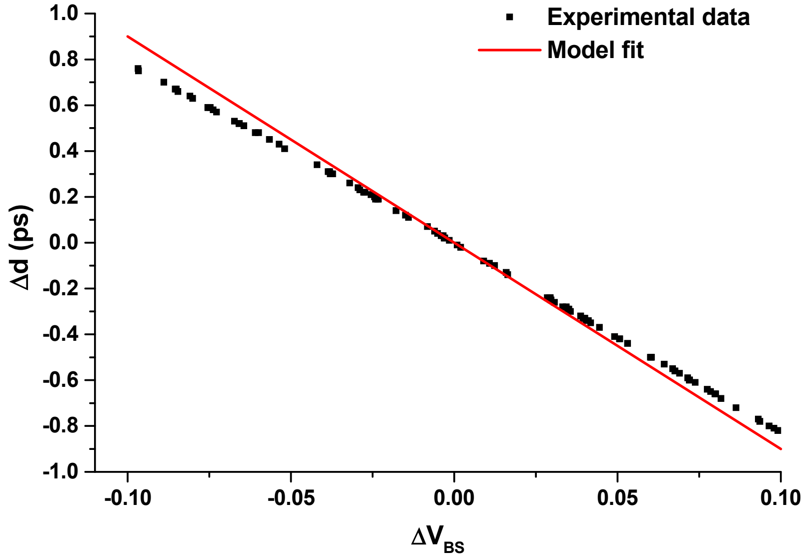


Figure 4.1: The dependence of delay on body bias.

concern, adaptive body bias may be the most useful tool. It has been demonstrated [40][42] that body biasing can be employed as an extremely effective knob to perform post silicon optimization and performance tuning by reducing the leakage for those dies that violate power constraints and increasing the frequency of those dies that do not meet delay specs.

The adaptive body bias technique exploits the dependency of the threshold voltage of a MOSFET device on its source-to-body voltage to achieve dynamic tuning of its delay and leakage power. For an NMOS device, the threshold voltage can be expressed as [33]:

$$V_{th} = V_{th0} + \gamma(\sqrt{V_{SB} + 2\phi_f} - \sqrt{2\phi_f}) \quad (4.2)$$

where V_{th0} is the threshold voltage of the device with zero body bias, γ is the body bias coefficient, and ϕ_f is the Fermi potential. Decreasing the source potential relative to the body of an N-channel device, translates to a negative V_{SB}, and decreases the threshold

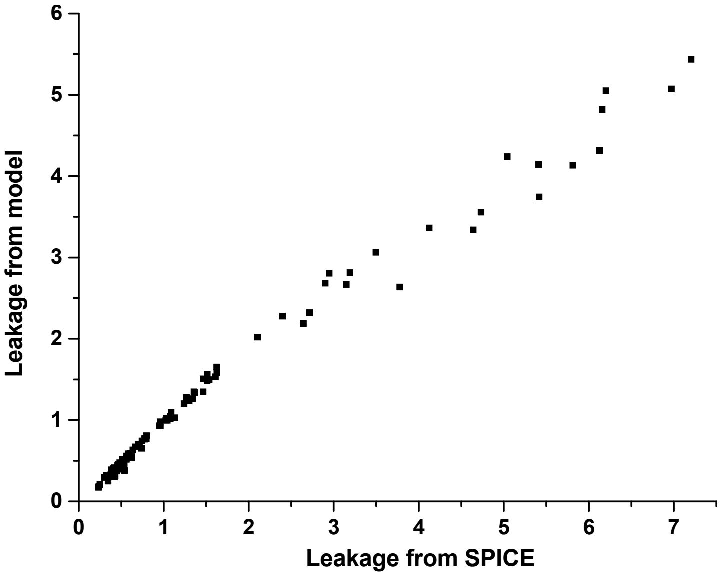


Figure 4.2: Comparison of the normalized leakage of inverter predicted by SPICE and the analytical leakage model

voltage. This technique, known as forward body biasing (FBB) reduces the delay of the gate at the expense of leakage power. On the other hand, application of reverse body bias (RBB) by applying a positive V_{SB} causes the threshold voltage of the device to increase. RBB is thus very effective in reducing the leakage power consumption.

For nominal delay, piecewise linear models are used.. The variability is assumed to come from two major sources. Transistor gate length (L) exhibits strong lithography induced variability. Threshold voltage (V_{th}) variation due to oxide thickness and dose variation is also taken into account. The impact of L on V_{th} due to drain-induced barrier lowering is predicted by the device model directly, which permits modeling L and V_{th} as independent random variables. Both L and V_{th} are assumed to follow the normal distribution. An additive statistical model that decomposes the variability, of both L and V_{th}, into the global (chip-to-chip) and local (intra-chip) uncorrelated variability components is used.

The impact of process parameter variability on gate delay is captured using a first-order parametric delay model:

$$\Delta d \cong S_1 \Delta L + S_2 \Delta V_{th} + S_3 \Delta V_{SB} \tag{4.3}$$

where ΔL and ΔV_{th} are the parameter deviations and ΔV_{SB} is the applied body bias. The sensitivities are the first-order derivatives of delay with respect to the specific variable (L, V_{th}, V_{SB}).

Using a modeling approach similar to [34], the subthreshold leakage current of a gate is expressed as an exponential function of the random parameters as:

$$I = I_o \cdot \exp(a\Delta L + b\Delta V_{th} + c\Delta V_{SB}) \tag{4.4}$$

where I_o is the nominal value of leakage per unit width. We obtain a good fit using this model (Figure 4.2), the *rms* error being ~8%. For a circuit block the expression for leakage can be expressed as:

$$I_{tot} = \sum_i \beta_i \cdot w_i \cdot \exp(a_i \Delta L_i + b_i \Delta V_{th,i} + c_i \Delta V_{SB}) \tag{4.5}$$

where the nominal gate leakage is $I_{0,i} = \beta_i \cdot w_i$.Following [34], we assume that the impact of random component of variation on chip-level leakage value can be captured by a constant multiplier that we take to modify the value of β_i, in the above expression.

4.1.2 Affine Model for Body Bias

The essence of adjustable optimization framework is that the variable that is allowed to be tuned is not determined arbitrarily but is dependent in some way on the realizations of uncertain variables. A computationally tractable solution to a statistical adjustable problem requires ΔV_{SB} to be an affine function of uncertain parameters, L and V_{th}:

$$\Delta V_{SB} = \pi_0 + \pi_1 \Delta L_g + \pi_2 \Delta V_{th,g} \tag{4.6}$$

The coefficients π_0, π_1 and π_2 are to be determined in the process of optimization. Such a parameterization is physically equivalent to compensating for the variation in leakage due to L and V_{th}, by applying body bias [23]. Though, the value of body bias is not a random variable, based on (4.6) it can be treated mathematically as one. With that observation, let us define $X_i = N(\mu_i, \sigma_i^2) = a_i \Delta L_g + b_i \Delta V_{th,g} + c_i \Delta V_{SB}$

The mean and variance of a lognormal $Y = e^X$ in terms of the mean and variance of the normal random variable $X = N(\mu, \sigma^2)$ are [58]:

$$\begin{aligned} E(Y) &= \exp(\mu + \frac{\sigma^2}{2}) \\ Var(Y) &= \exp[2(\mu + \sigma^2)] - \exp(2\mu + \sigma^2) \end{aligned} \tag{4.7}$$

Observing that $E(I_{tot}) = \sum_i E(\beta_i \cdot w_i \cdot \exp(a_i \Delta L_g + b_i \Delta V_{th,g} + c_i \Delta V_{SB}))$ and $\mu_i = E(a_i \Delta L_g + b_i \Delta V_{th,g} + c_i \Delta V_{SB}) = \pi_o$ we can write the expected value of total block leakage as:

$$E(I_{tot}) = \sum_i \beta_i \cdot w_i \cdot \exp(\pi_o + \sigma_i^2 / 2) \tag{4.8}$$

4.2 DESIGN TIME / POST SILICON CO-OPTIMIZATION USING ADAPTABLE ROBUST OPTIMIZATION

In the optimization strategy we develop, the optimal body bias is determined after the realization of uncertainty of the process parameters. On-chip measurements are used to measure the actual parameter values and their deviations from nominal values. Then, the policy derived during optimization can be used to choose an optimal forward or reverse body bias. RBB can be applied to reduce yield loss in the high frequency (high leakage bins), and can be used FBB to tighten the distribution at the low frequency bins.

4.2.1 Adaptable Robust Optimization

First we introduce the theoretical foundation for robust adjustable optimization. We use robust optimization as the bedrock of our strategy. A robust LP can be defined as the problem of minimizing the worst-case of a linear objective and constraint functions [59]:

$$\min\{ \sup_{\zeta \equiv [A,b,c] \in Z} (c^T x) : Ax \le b \ \ \forall \zeta \equiv [A,b,c] \in Z\} \tag{4.9}$$

Here the uncertainty in the matrix coefficients is represented as $\zeta \equiv [A,b,c]$ varying in the nonempty compact convex uncertainty set Z.

The above problem requires all decisions to be made prior to the actual realization of the uncertain parameters. However, in many real-life cases not all the decisions have to be made simultaneously: only some variables may become known earlier. In that case, the remaining decision variables can be adjusted to the realizations of uncertain data. It is obvious that if the opportunity to adjusting some variables is given, the optimal solution will be better (or at least, no worse) than for the problem above. Problems with similar structure have been known as multi-stage stochastic problems with recourse, and are known to be intractable. Robust problems are not stochastic problems, and can be solved in polynomial time. We can re-write the problem of (4.9) in terms of the non-adjustable variables u and the adjustable variables v. Then, the adjustable robust problem can be defined as:

$$\min\{c^T \begin{pmatrix} u \\ v \end{pmatrix} : \forall (\zeta \equiv [U,V,b,c] \in Z) \ \exists v : Uu + Vv \le b\} \tag{4.10}$$

In this formulation, the adjustable variables v are allowed to depend on the realization of ζ.

Still, it is shown in [60] that the general robust problem with adjustable parameters is NP-complete, unless restrictions are applied on how exactly the adjustable variables tune themselves to uncertain data. It is also shown that a computationally feasible adjustable robust linear problem can be achieved if *the adjustable variables are constrained to be affine functions of the uncertain variables*. This is equivalent to:

$$v = w + W\zeta \tag{4.11}$$

From this we see that the adjustable variables can be tuned once the realization of uncertain data is known. However, if we are to be able to identify an optimal policy and do that computationally efficiently, the dependency cannot have general form, but must be constrained. This ultimately leads to the affinely adjustable robust linear program:

$$\min\{c^T \begin{pmatrix} u \\ v \end{pmatrix} : Uu + V(w + W\zeta)v \le b \ \forall \zeta \equiv [U, V, b, c] \in Z\} \tag{4.12}$$

In particular, for uncertainty sets specified using linear or second-order cone constraints, the above problem can be reformulated as an LP or a second-order conic program [60].

4.2.2 Co-Optimization: Problem Formulation

We now map our design-time and post-silicon tuning problem into a robust adjustable linear program. Our objective in formulating the problem is to set up a robust linear program with adjustable parameters.

The task of co-optimization is effectively a two-stage optimization problem with recourse. Denoting column vectors by boldface letters, we formulate the problem as that of minimizing the overall expected leakage power (or current) with expectation being taken over the population of manufactured chips while satisfying timing constraints under a statistical timing model:

$$\min E(I_{tot}) \quad s.t.\, P(D(\boldsymbol{w}, \Delta V_{SB}) \le T) \ge \alpha \tag{4.13}$$

In this formulation the objective and constraint functions are dependent both on design-time variables (gate sizes) and post-silicon optimization variables (ΔV_{SB}). We begin by writing the expression for mean leakage as:

$$E(I_{tot}) = \boldsymbol{g}^T\boldsymbol{w} \tag{4.14}$$

where $\boldsymbol{g}$ is an $N \times 1$ vector with entries $g_i = \beta_i \exp(\pi_o + \sigma_i^2 / 2)$. The objective function is thus linear in the gate sizes and non-linear (exponential) in ΔV_{SB}. We will deal with this by adopting a linearization approach in which we locally linearize the objective's dependence on ΔV_{SB} at the fixed value of vector of gate sizes $\boldsymbol{w}$.

Let us, for convenience form a single vector of decision variables $\boldsymbol{x} = [\boldsymbol{w}\,\Delta V_{SB}]^T$. The gate delay model introduced in the previous section can allow us to express path timing constraints in the form of:

$$D(\boldsymbol{w}, \Delta V_{SB}) = a^T x \tag{4.15}$$

Now consider the probabilistic chance constraint $P(D(\boldsymbol{w}, \Delta V_{SB}) \le T) \ge \alpha$ specified for the entire circuit. We can heuristically re-write the circuit-level probabilistic timing constraints in terms of path-based constraints as discussed in Chapter 2. We assume that a corresponding confidence level η_j can be selected. Then, we require:

$$P(D_i(\boldsymbol{w}, \Delta V_{SB}) < T) \ge \eta_j \quad \forall j \in \Pi \tag{4.16}$$

where Π is the relevant path-set. Relying on the linear vector representation introduced above we can write:

$$P(D_i(\boldsymbol{w}, \Delta V_{SB}) \le T) = P(\boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{x} \le T\,) \ge \eta_j \, \forall j \in \Pi \tag{4.17}$$

If $\boldsymbol{a}$ is distributed normally, $N(\overline{a}, \Sigma)$, the coefficients of $\boldsymbol{x}$ belong to an ellipsoidal uncertainty set. Then, it can be shown that the above constraint is equivalent to:

$$\overline{\boldsymbol{a}}^T\boldsymbol{x} + k_j(\boldsymbol{x}^T\Sigma\boldsymbol{x})^{1/2} \le T \quad \forall j \in \Pi \tag{4.18}$$

where $k_j = \phi^{-1}(\eta_j)$ and ϕ is the cumulative distribution function (*cdf*) of the standard normal distribution. The path delay constraints of (4.18) represent a set of second-order conic path timing constraints.

It has been shown that adjustable robust linear programs can be made computationally tractable only if the adjustable (second- stage) decision variables are *affine* functions of uncertain variables [60]. Without loss of generality, consider only the global sources of variation ΔL_g and $\Delta V_{th,g}$, and a single value of body bias ΔV_{SB} for all the gates on the chip. Then, the affine policy is given by:

$$\Delta V_{SB} = \pi_0 + \pi_1 \Delta L_g + \pi_2 \Delta V_{th,g} \tag{4.19}$$

This dependence can be used to express the expected value of leakage current as:

$$g_i = \beta_i \exp\left(f_{0,i}^2(\pi_0) + f_{1,i}^2(\pi_1)\sigma_{L_g}^2 + f_{2,i}^2(\pi_2)\sigma_{V_{th,g}}^2 \right) \tag{4.20}$$

where $f_{0,i}$, $f_{1,i}$, and $f_{2,i}$ are linear functions of π_0, π_1, and π_2 respectively.

The robust adjustable optimization problem can now be formulated as:

$$\begin{aligned} &\min \ \boldsymbol{g}^T \boldsymbol{w} \\ &\bar{\boldsymbol{a}}_j^T \boldsymbol{x} + \phi^{-1}(\alpha_j)(\boldsymbol{x}^T \Sigma_j \boldsymbol{x})^{1/2} \le T \quad \forall j \in \Pi \end{aligned} \tag{4.21}$$

where $g_i = \beta_i \exp\left(f_{0,i}^2(\pi_0) + f_{1,i}^2(\pi_1)\sigma_{L_g}^2 + f_{2,i}^2(\pi_2)\sigma_{V_{th,g}}^2 \right) \ \forall i \in [1, N]$Note that the original problem has now been cast as an optimization problem in π_0, π_1, and π_2 and gate widths, w_i. The solution to this problem is an optimal policy $P = (\pi_0, \pi_1, \pi_2)$ and the vector of gate width $\boldsymbol{w}$ such that the timing constraints are satisfied.

4.3 PROBLEM SOLUTION

To enable a computationally efficient solution, we solve the problem as a two phase optimization program. The first phase consists of solving a weighted sizing problem assuming fixed body bias and the second phase consists of solving for the body bias value assuming fixed gate size. This is performed in an iterative manner using successive approximations until the solution converges.

Denoting column vectors using boldface letters, the final optimization problem $ABB(\boldsymbol{w}, \boldsymbol{\pi})$ can now be expressed as:

$$\begin{aligned} &\min \ \boldsymbol{g}^T \boldsymbol{w} \\ &\bar{\boldsymbol{a}}_j^T \boldsymbol{x} + \phi^{-1}(\alpha_j)(\boldsymbol{x}^T \Sigma_j \boldsymbol{x})^{1/2} \le T \quad \forall j \in \Pi \end{aligned} \tag{4.22}$$

where $g_i = \beta_i \exp\left(f_{0,i}^2(\pi_0) + f_{1,i}^2(\pi_1)\sigma_{L_g}^2 + f_{2,i}^2(\pi_2)\sigma_{V_{th,g}}^2 \right) \ \forall i \in [1, N]$ Here $\boldsymbol{\pi}$ denotes the vector of πs. We transform the path based formulation into a node based formulation [26] to solve the problem efficiently.

This problem is solved iteratively by computing optimal ws in the first stage and optimal πs in the second stage until the solution converges. At an iteration l the w-phase consists of solving $ABB(\boldsymbol{w}, \boldsymbol{\pi}^{(l-1)})$ to obtain $\boldsymbol{w}^{(l)}$ and the π-phase solves the problem $ABB(\boldsymbol{w}^{(l)}, \boldsymbol{\pi})$ to obtain $\boldsymbol{\pi}^{(l)}$. Initially, for $l = 0, \pi_j = 0 \ \forall j \in [1, k]$ corresponding to zero body bias.

Solving w-phase does not pose a problem as the objective function is linear in gate widths, $\boldsymbol{w}$ and the delay constraints are second order cones. It can therefore be solved readily as an SOCP. However, the π-phase objective is non-linear in the decision variables. To address this issue we propose to expand the objective function using a first order Taylor series. The π- phase optimization problem solved at iteration l is approximated as:

$$\begin{aligned} &\min F \\ &s.t. F \ge \pi_0 \nabla_{\pi_0}(\boldsymbol{g}^T \boldsymbol{w}) + \pi_1 \nabla_{\pi_1}(\boldsymbol{g}^T \boldsymbol{w}) + \pi_2 \nabla_{\pi_2}(\boldsymbol{g}^T \boldsymbol{w}) \\ &\bar{\boldsymbol{a}}_j^T \boldsymbol{x} + \phi^{-1}(\alpha_j)(\boldsymbol{x}^T \Sigma_j \boldsymbol{x})^{1/2} \le T \quad \forall j \in Paths \end{aligned} \tag{4.23}$$

where ∇_{π_0}, ∇_{π_1} and ∇_{π_2} are the gradients computed w.r.t π_0, π_1, and π_2 respectively. The complete algorithm **optim_abb** is presented in Figure 4.3.

```
1. set $\pi_i = 0 \ \forall i \in [1,k]$
2. get Timing Target $T$
3. set $D < T$ such that $ABB(\boldsymbol{w}, \boldsymbol{\pi}^{(l-1)})$ is feasible.
4. chose delay increment $\delta D$
5. set $l = 1$
6. if $D < T$
       solve $w$ - phase $ABB(\boldsymbol{w}, \boldsymbol{\pi}^{(l-1)})$ setting delay
       constraint to $D$ .
   else
       print $\boldsymbol{w}^{(l-1)}$ and $\boldsymbol{\pi}^{(l-1)}$ as the optimal solution and
       stop
7. set $D = D + \delta D$
8. if $D < T$
       solve $\pi$ -phase $ABB(\boldsymbol{w}^{(l)}, \boldsymbol{\pi})$ setting delay constraint
       to $D$ .
   else
       print $\boldsymbol{w}^{(l)}$ and $\boldsymbol{\pi}^{(l-1)}$ as the optimal solution and stop
9. set $D = D + \delta D$
10. set $l = l + 1$ and goto step 6
```

Figure 4.3: The two phase algorithm optim_abb for post silicon optimization using ABB.

4.3.1 Accounting for Intra-chip Variation

The policy described above cannot account for random parameter variation. Since the structure of the policy needs to be specified at optimization time, we need to know the number of measurements we can make on chip to account for the intra-chip random variation. Assume that we can make k_l measurements L and k_v measurements of V_{th}. Assuming that we are allowed a single choice of ΔV_{SB}:

$$\Delta V_{SB} = \pi_0 + \sum_{i=1}^{k_l} \pi_i \Delta L_i + \sum_{i=1}^{k_v} \pi_{k_l+i} \Delta V_{th,i} \tag{4.24}$$

The notion of measurement complexity $k = k_l + k_v$ is used here to represent the amount of information we are able to obtain about the structure of variability. As we

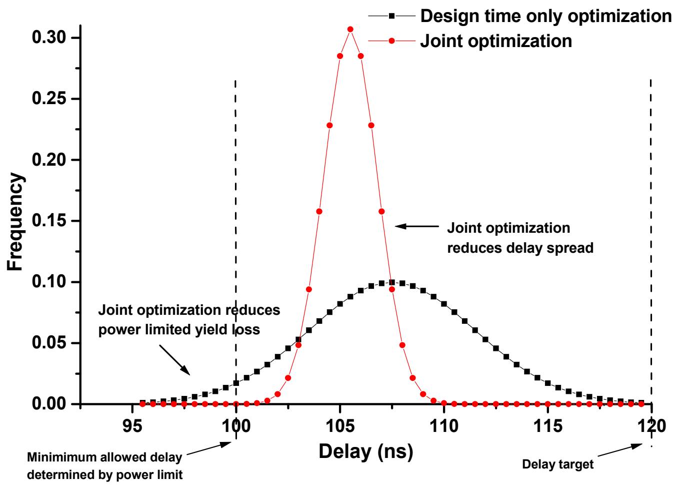


Figure 4.4: PDFs of delay distributions produced by design- time only optimization and joint optimization.

demonstrate in the results section, a higher value of k implies a lower leakage value. However, it is achieved at the cost of increased run-time and diagnostic overhead.

Similarly, we can introduce the notion of control complexity n which refers to the number of body bias values that are allowed. Control complexity reflects the degree of controllability over the body bias assignment and also the circuit overhead. It is currently assumed that the granularity of body bias assignment is at the block level. This is because tuning individual gates is clearly too expensive from the physical design perspective (extra routing overhead, voltage conversion). Spatial clustering may also be used as the gates that are spatially proximate are more likely the benefit from an equal body bias assignment.

4.4 EXPERIMENTAL RESULTS

We are now in a position to put together the complete design-time and post silicon co-optimization flow. We start by choosing the level of measurement complexity and control complexity. These along with the distributional information about the uncertain

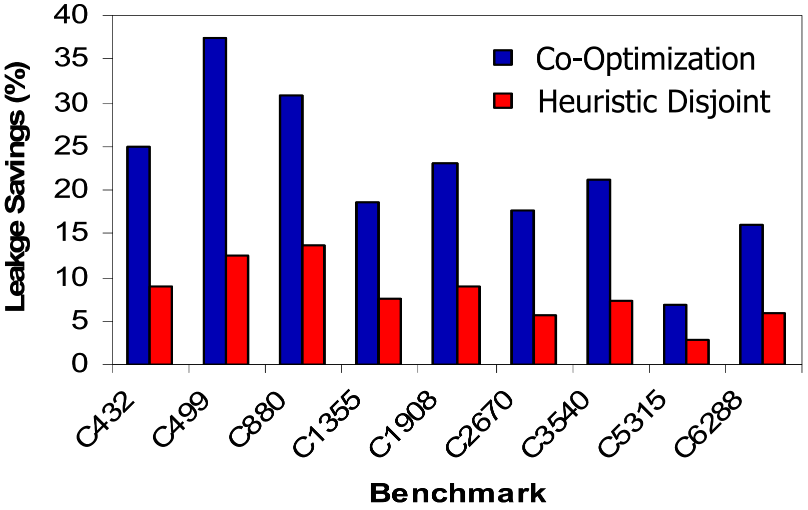


Figure 4.5: Comparison of disjoint optimization and joint design time and post silicon optimization.

data are the inputs to the above algorithm. The algorithm optim_abb produces a set of gate sizes and an optimal policy for selecting ΔV_{SB} for the given structure of variation and the control and measurement complexity. When the chip is fabricated, the actual realizations of the uncertainty are known hence the value of body bias is determined from the policy from (4.24).

The optimization problem was solved using the conic optimization package MOSEK [57]. The experiments were run on a 32-bit, 3.7 GHz. Intel Xeon processor with 4GB of memory. The benchmark circuits were synthesized to a cell library that was characterized for a 70 nm process using Berkeley Predictive Technology Model. For NMOS (PMOS) transistors, the threshold voltage is 0.10V (-0.10V). The assumed magnitude of V_{th} and L variability is $\sigma / \mu = 8\%$ and 5% respectively. The optimal solution (sizes and policy) produced by the algorithm were evaluated using Monte Carlo analysis to estimate the expected value of leakage power by sampling from the distribution of the uncertain parameters V_{th} and L.

Figure 4.4 illustrates the effectiveness of our algorithm in reducing the spread of the circuit delay and ameliorating the problem of the dual ended squeeze on parametric yield. This is achieved by increasing the delay of faster chips by applying RBB. Since these chips have high leakage power consumption, our algorithm reduces power limited yield loss. From the Figure it can be seen that the yield is improved by about 5%.

We performed comparison against heuristics that performs post silicon tuning separately after sizing. Since it is difficult to pick optimal value of body bias for cells in design, disjoint tuning performs worse than joint optimization: the delay spread is higher and the leakage power consumption is greater (Figure 4.5).

Three measures of complexity are used to characterize the optimality of the solution: the control complexity n which represents the granularity of control, the measurement complexity k which refers to the granularity of the monitoring and sensing circuitry, and the parameter complexity ρ, defined as the ratio $\sigma_l^2 / \sigma_{tot}^2$. Thus, ρ is a measure of how spatially uncorrelated the process variable is.

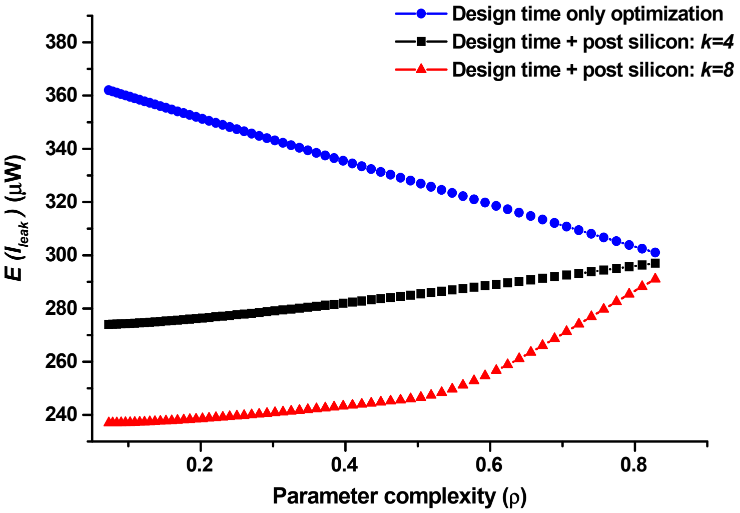


Figure 4.6: Comparison of design time only optimization and joint design time and post silicon optimization.

Application of FBB to slow chips serves to tighten the delay distribution further. Since the circuit is guaranteed to meet the timing yield target even for zero FBB, applying forward body bias does not improve timing yield but increases the number of chips in the higher frequency bins. Figure 4.6 compares the leakage power of the circuits obtained by employing only design time optimization and the joint design time and post silicon algorithm outlined in the paper. As expected, using post silicon optimization enables a more optimal solution compared to design time only optimization. However as the complexity of variability increases, the benefit of using post silicon optimization decreases. This can be attributed to the fact that as the amount of uncorrelated variability increases, design time optimization performs better, but to utilize the adaptability provided by post silicon optimization, more measurements need to be made and more complex control system used (larger number of individually clusters of logic on a chip). Therefore, increasing measurement complexity k improves the quality of the solution (reduces expected value of leakage). This is also depicted in Figure 4.7 However, this

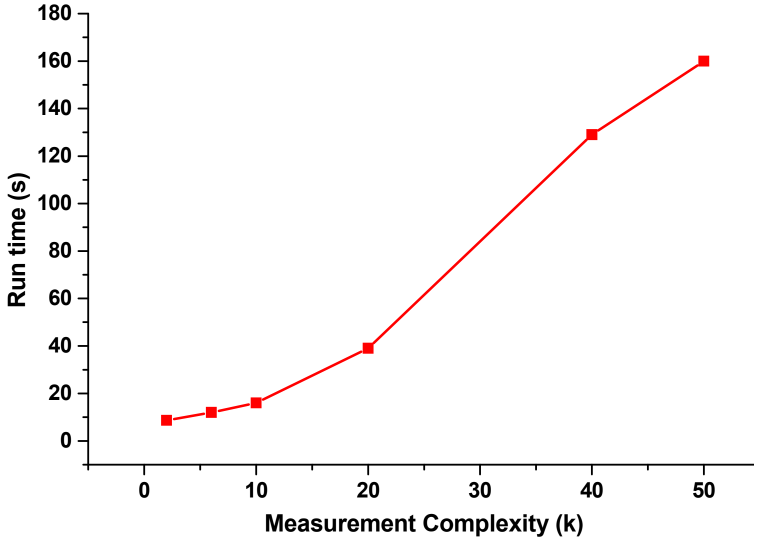


Figure 4.7: The runtime increases as the measurement complexity is increased, as optimal policy depends on more measurements.

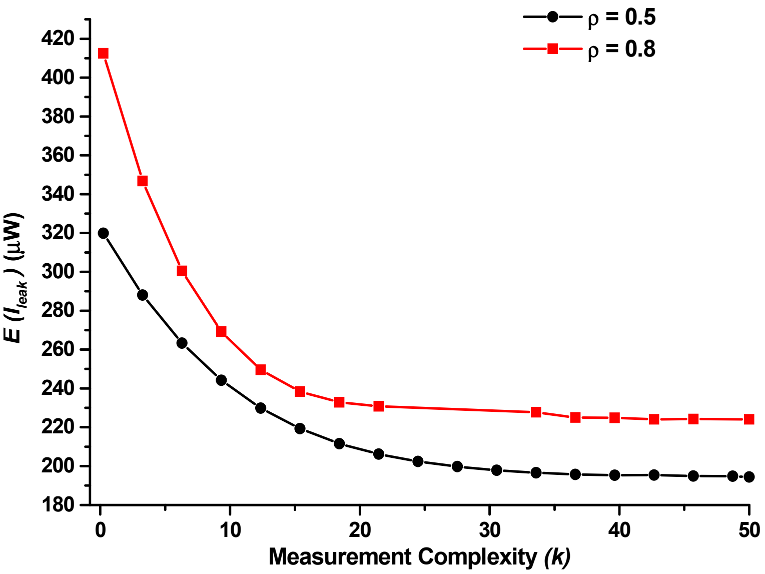


Figure 4.8: The expected value of leakage power decreases as we increase measurement complexity but the benefits level off for high values of *k*.

comes at the cost of increased run-time and diagnostic overhead. This is shown in Figure 4.7, which indicates that the run-time of the algorithm increases as k is increased.

Table 4.1 documents the results obtained across the benchmarks. All solutions were evaluated using Monte Carlo analysis. 1000 samples were generated for each random parameter. The circuits were optimized for the same delay target, which is evaluated using Monte Carlo. We observe that for a reasonable choice of measurement complexity, using our algorithm, an average saving of 20% savings in leakage power consumption can be obtained compared to design time only optimization. Table 5.1 also cites the runtimes of the algorithm. It can be seen that the run time behavior is extremely good (about 2 minutes) even for the largest benchmark circuit.

Finally, we explore the dependence of the quality of the solution obtained from post silicon optimization on the measurement complexity and control complexity. Increasing k improves the leakage power but there is a point of diminishing returns beyond which the improvement is insignificant. This is depicted in Figure 4.8. Increasing the number of circuit clusters with individually adjustable threshold voltages (i.e., increasing the control complexity) improves the results of optimization, Figure 4.9. As with measurement complexity, this improvement in leakage power is achieved at a cost. A larger value for control complexity implies greater overhead, such as in biasing circuitry and routing.

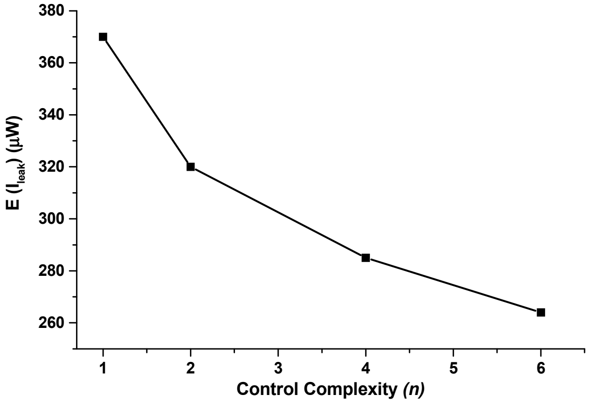


Figure 4.9: The expected value of leakage power decreases as we increase control complexity *n*. A larger *n* corresponds to more allowable values of body bias.

Table 4.1: Leakage power savings obtained by the joint post-silicon and design-time optimization.

Circuit	No. of gates	Design time optimization $E(I_{leak})$ (μW)		Joint design-time and post-silicon optimization ($k = 8$)				
		$\rho = 0.5$	$\rho = 0.8$	$E(I_{leak})$ (μW)		Leakage power savings (%)		Runtime (s)
				$\rho = 0.5$	$\rho = 0.8$	$\rho = 0.5$	$\rho = 0.8$	
C432	261	328	301	246	291	25.00	3.32	8
C499	641	908	845	568	622	37.44	26.39	15.2
C880	615	560	470	388	405	30.71	13.83	12.5
C1355	685	684	603	557	595	18.57	1.33	21.1
C1908	1238	1203	1167	926	1040	23.03	10.88	31
C2670	2041	1706	1669	1405	1530	17.64	8.33	55
C3540	2582	2718	2584	2142	2473	21.19	4.30	63
C5315	3753	3801	3700	3544	3598	6.76	2.76	108
C6288	2704	2918	2902	2454	2685	15.90	7.48	132
Average savings						**21.8**	**8.73**	

4.5 SUMMARY

In this chapter we have developed a theoretical foundation for joint design-time and post-silicon optimization. The problem is cast as an adjustable robust linear program and solved in a computationally efficient way. Results indicate that the designer can greatly benefit from synergistic application of design time and post silicon optimization techniques due to the ability of post silicon optimization solution to tune itself to the realization of uncertain data and up to 20% savings in leakage power can be obtained. We have also introduced metrics that enable designers to assess the complexity of biasing circuitry needed. In the next chapter we explore he notion of flexible tuning in an application with limited second stage adaptability using the framework of finite adaptable optimization.

Chapter 5: Design of Power-Optimal Buffers Tunable to Process Variability

Given a capacitive load C_L, the problem of buffer design is to find the values of the sizing factors such that the required objective function is minimized. The problem can be formulated using several objectives including propagation delay through the buffer chain or its energy-delay product [43]. The set of optimal designs can be represented by the Pareto curve in the energy-delay space [52] . For some cases, the optimization above can be done analytically. For example, it can be shown that a minimum delay buffer chain can be designed by setting the taper factor to 2.7.

In the presence of variability, conventional buffer design principles may not be most effective. Indeed, guaranteeing that timing constraints are met even under the "slow" process conditions requires over-designing the buffer, in the sense that for most instances of the process the area and power could be smaller. The optimization challenge that we address is how to choose the values of parameters that are fixed during the design and how to select between the alternatives after manufacture such that the timing yield is met, and average power is minimized. The problem is formulated statistically in terms of yield of timing and power metrics. We seek to minimize the total power of the buffer, under a delay constraint T at a given yield γ. Thus we have:

$$\min\ P_{dyn} + E(P_{leak})\ s.t.\, P(D \le T) \ge \gamma \tag{5.1}$$

Here, P_{dyn} is the dynamic power consumption and P_{leak} the leakage power. Here, $P()$ denotes the probability measure of the variability. This formulation is general, and allows us to explore various power, delay and yield trade-offs.

5.1 ADAPTABLE BUFFER DESIGN PRINCIPLES

Variability is assumed to be due to two major sources- transistor gate length (L) and threshold voltage. Because of the relatively small size of buffers compared to chip size, only global variability in L and V_{th} is considered. The impact of process parameter variability on gate delay is captured using a first-order model. The deviation from nominal delay of the buffer is

$$\Delta D = \sum_{j=1}^{N} (S'_{1_j} \Delta L + S'_{2_j} \Delta V_{th}) \tag{5.2}$$

where the sensitivities S'_{1_j} and S'_{2_j} of gate j have a posynomial dependence on its size and load. The sub-threshold leakage power of a gate is expressed as an exponential function of the random parameters. For a buffer chain the expression for leakage can be written as:

$$P_{leak,chain} = \sum_{i \in chain} (1 - \alpha_{act,i}) P_{0,i} w_i \exp(a_i \Delta L + b_i \Delta V_{th}) \tag{5.3}$$

5.1.1 Overview of Finite Adaptable Optimization

Adaptable optimization addresses optimization with uncertain parameters, where there are two stages of decisions (in our case, the first stage is the design phase, and the second is the post-silicon tuning phase). Second stage decisions depend on the realization of the uncertainty. Adaptable optimization seeks to make optimal design decisions in the first phase, given that such tuning is possible. Post-silicon tuning options are limited by physical constraints, as well as by the expense and implementation overhead of having to solve very complex post-silicon tuning optimization problems for every chip. Finite adaptable optimization [71] is a framework intended for such limited second-stage adaptability. The second stage decisions are discrete, and chosen from a finite set of pre-determined "contingency" plans. We denote the realization of the uncertainty as ω, and the uncertainty set in which ω takes values as Ω. A partition of the uncertainty set Ω

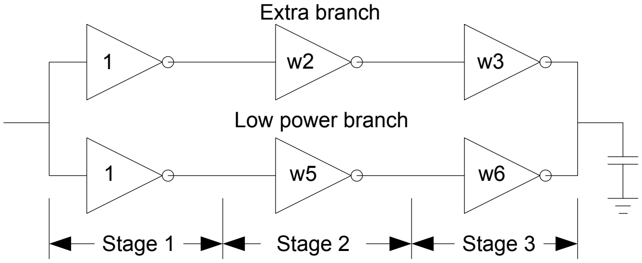


Figure 5.1: Conceptual representation of the adaptable buffer

into k regions is selected, such that $\Omega = \bigcup_{i=1}^{k} \Omega_i$. Depending on the region the realized uncertainty falls in, a contingency plan is selected for the second stage. The first-stage design, and the contingency plans are optimally selected by solving:

$$\min_{\Omega = \bigcup_{i=1}^{k} \Omega_i} \begin{bmatrix} \min c^T x + \max\{d^T y_1, \ldots, d^T y_k\} \\ s.t. A(\omega)x + B(\omega)y_i \geq b, \ \forall \omega \in \Omega_i, \ i \in [1,k] \end{bmatrix} \tag{5.4}$$

5.1.2 Buffer Design as a Finite Adaptable Optimization Problem

In the probabilistic setting which we consider below, the max can be replaced by an expected value. We use this framework as our optimization strategy to design a small number of alternative buffers (our contingency plans). The alternatives are due to the buffer chain that has a low power branch and an additional high-speed branch (Figure 5.1). In the post-silicon phase, a buffer is selected depending on the realization of uncertain variables: e.g. if gate length is smaller than a pre-determined quantity, we use the low-power version of the buffer, if it is above, we use the high-speed version. The result is flexible and effective post silicon tuning with minimal implementation overhead. This set of individual buffers is henceforth collectively referred to as a ‘buffer configuration’. The sizing factors for common inverters correspond to the first stage decision variables. The second stage variables are the non-common portions in the buffers, and only these depend on ω. Consistent with other approaches, the sizing factor of the first inverter is chosen to be unity.

Our objective is to optimize the common inverters and the multiple buffers such that the total power for the entire configuration is minimized. For the sake of convenience, let us represent the objective function $P_{dyn} + E(P_{leak})$ by P_{tot}. For a design with k alternative buffers, this corresponds to a partition of the uncertainty set into k regions. Thus we adapt (5.4) to get:

$$\begin{aligned} &\min_{\Omega=\bigcup_{j=1}^{k}\Omega_i} \sum_{\mathrm{j}=1}^{\mathrm{k}} P_{tot}(\boldsymbol{x},\boldsymbol{y_j},\omega)P(\omega \in \Omega_j) \\ &s.t.\, P(D_j(\boldsymbol{x},\boldsymbol{y_j},\omega) \le T \mid \omega \in \Omega_j) \ge \gamma, \forall j \in [1,k] \end{aligned} \tag{5.5}$$

Here, the vector $\boldsymbol{x}$ represents the sizes of the inverters in the common part of the buffer configuration and $\boldsymbol{y}_j$ the non-common portion for buffer j. In (5.5), the objective and constraints have the same coefficients for the first and second stage variables ($\boldsymbol{c} = \boldsymbol{d}, \boldsymbol{A} = \boldsymbol{B}$). When there is no shared portion between the buffers in a configuration, the problem decouples, and can easily be parallelized [71].

5.1.3 Design of Adaptable Buffer

The adaptable buffer is designed using tri-state inverters as shown in Figure 5.2. The high-speed branch is turned on using the *ctrl* signal. The delay and power of the high-speed and low-power buffers are modeled as posynomial and linear functions of the inverter widths respectively. The delay reduction in the high-speed case occurs due to the sharing of the load between the two inverters in the final stages of the two chains.

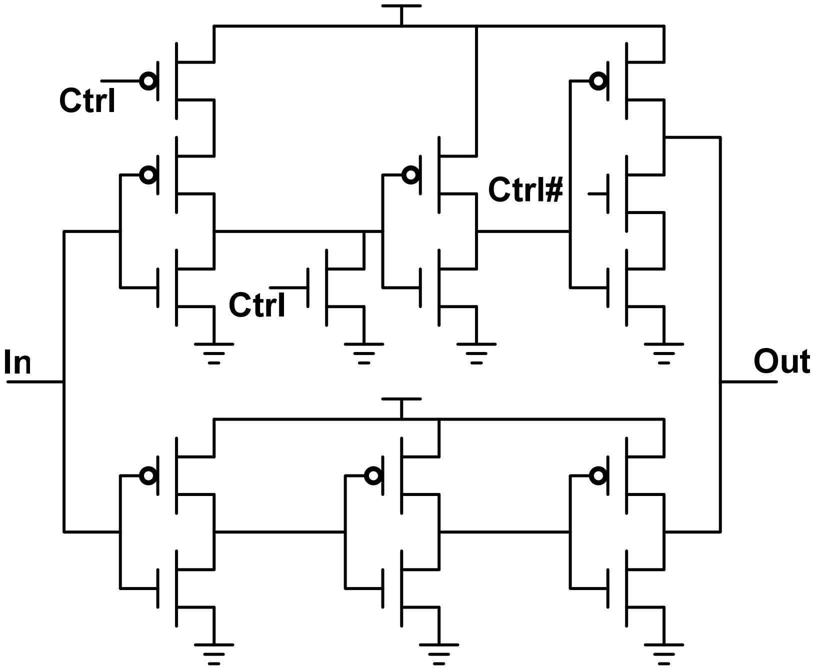


Figure 5.2: Adaptable buffer design using tristate inverters

5.2 BUFFER OPTIMIZATION STRATEGY

In this section the optimization strategy is described in detail. The objective is to co-optimize design time and post-silicon tunability in a mathematically rigorous manner, to obtain the first and second stage decisions (sizes or body biases) of the common and non-common portions of the buffers, and the optimal partition of the uncertainty set.

5.2.1 Partitioning the Uncertainty Set

In the initial presentation, we restrict our attention to only one varying process parameter, the effective channel length L, and partition of the uncertainty into only two pieces. This is done merely to explain the procedure more clearly, and the full treatment is shown later. Though multiple partitions could be considered, this carries the penalty of increased area and control circuitry cost.

In this initial setting, we must select a truncation point l_0, and design two buffers, one to be used if the channel length realization satisfies $\tilde{L} \in [l_0, \infty)$, and the other for $\tilde{L} \in (-\infty, l_0]$. The objective of the optimization is to find the sizes of the inverters in the buffers and the truncation point l_0 which defines the decision policy.

For the sake of presentation let $x \in \left\{x_1, x_2\right\}$ be a selector variable that refers to whether buffer 1 or 2 is selected. Assuming $L \sim N(\mu_L, {\sigma_L}^2)$, for any truncation point l_0, let α_1 denote the probability of selecting the low power buffer and α_2 denote the probability of selecting the high speed buffer . (Henceforth, the term truncation point is used interchangeably to refer to α_1 or l_0.) It follows that

$$\begin{aligned} \alpha_1 &= \mathrm{P}(x = x_1) = P(L \le l_0) = \phi(\frac{l_0 - \mu_L}{\sigma_L}) \\ \alpha_2 &= \mathrm{P}(x = x_1) = 1 - \alpha_1 \end{aligned} \tag{5.6}$$

where ϕ is the *cdf* of $N(0,1)$. Define random variables L_1 and L_2 with ranges $(-\infty, l_0]$ and $[l_0, \infty)$ whose *pdf* and *cdf* is the rescaled conditional *pdf* of L, conditioned on being smaller or larger, respectively, than l_0.

5.2.2 Formulating the Constraints and Objective

Let D denote the delay through the buffer configuration. Letting $D_i = (D \mid x = x_i), i = 1,2$, the delay constraint can be written as:

$$\begin{aligned} & P(D \le T) \ge \gamma \\ \Leftrightarrow\ & P(D_1 \le T)P(x = x_1) + P(D_2 \le T)P(x = x_2) \ge \gamma \end{aligned} \tag{5.7}$$

If $P(D_i \le T) = \beta_i, i = 1,2$, (5.7) reduces to

$$\alpha_1\beta_1 + \alpha_2\beta_2 \ge \gamma \;\; s.t. P(D_1 \le T) \ge \beta_1, P(D_2 \le T) \ge \beta_2 \tag{5.8}$$

The delay constraints are posynomial in the buffer sizes and linear in the process parameters, implying that delay is monotonic in the process parameters. This can be used to show that the following conditions are equivalent.

$$\begin{aligned}&P(D_i \leq T) \geq \beta_i \\ &\Leftrightarrow D(L_i) \leq T \,\forall\, L_i \leq l_i^* \Leftrightarrow D(l_i^*) \leq T \text{ where } l_i^* = F_{L_i}^{-1}(\beta_i)\end{aligned} \tag{5.9}$$

The objective function is the total power of the configuration. Let $P_{tot,config}$ denote the total power of the buffer configuration. Let $P_{tot,i} = (P_{tot} \mid x = x_i), i = 1,2$. The objective can be formulated as

$$\begin{aligned}&P(x = x_1)(P_{dyn,1} + E[P_{leak,1}]) + P(x = x_2)(P_{dyn,2} + E[P_{leak,2}]) \\ &= \alpha_1 P_{tot,1} + \alpha_2 P_{tot,2}\end{aligned} \tag{5.10}$$

Adopting an exponential leakage power model with $\Delta V_t = 0$, for a single gate we then have:

$$E(P_{leak,i}) = P_{o,i} w_i E(\exp(a\Delta L)) \tag{5.11}$$

where $\Delta L = L - \mu_L$. Total dynamic power of a buffer depends on buffer sizes and can be computed straightforwardly as the sum of the dynamic powers of the individual components. Consider the problem of computing the expected value of leakage power of a buffer.

$$E(P_{leak,i}) = \sum_{j \in i} E(P_{leak,j}) = \sum_{j \in i} P_{0,j} w_j E(\exp(a_1 \Delta L_i)) \text{ for } i = 1,2 \tag{5.12}$$

Observing that the term $\exp(a_1 \Delta L_i)$ is the moment generating function of ΔL_i for $t = a_1$, $E(P_{leak,1})$ can be expressed as:

$$E(P_{leak,1}) = \frac{F(l_0)}{\phi(l_0)} \exp(\sigma_L a^2 / 2) \sum_{j \in 1} P_{0,j} w_j \tag{5.13}$$

where ϕ is the *cdf* of $N(0,1)$ and $F(l_0)$ is the *cdf* of $N(\sigma_L^2 t, \sigma_L^2)$ (and similarly for $E(P_{leak,2})$). Using (5.13), the final optimization formulation can be stated as

$$\begin{aligned}&\min\ \alpha_1 P_{tot,1} + \alpha_2 P_{tot,2} \\ &D(l_1^*) \leq T,\ D(l_2^*) \leq T \text{ where } l_1^* = F_{L_1}^{-1}(\beta_1), l_2^* = F_{L_2}^{-1}(\beta_2)\end{aligned} \tag{5.14}$$

This is a geometric program (GP) for a given truncation point.

5.3 PROBLEM SOLUTION USING GEOMETRIC PROGRAMMING

We can now proceed to solve the adaptable buffer sizing problem. Given a delay constraint T timing yield γ and capacitive load C_L the first step is to pick the number of stages N. To obtain N, the non-adaptive formulation in (5.1) is solved repeatedly to find the smallest N for which the problem is feasible. This is also the smallest value of N for which the adaptable problem (5.14) is feasible. Since adding additional stages increases power consumption (Figure 5.5), this is the optimal value of N for the adaptable problem.

To obtain the optimal truncation point, we observe that the problem is quasi-convex in l_0. The optimal truncation point can be obtained quickly by bisection, although for simplicity we solve repeatedly via a linear sweep over the interval [0.0013,0.9987], corresponding to $\frac{l_0 - \mu_L}{\sigma_L} = [-3,3]$ (justified by quasi-convexity, and the results of Section 5.4). The overall algorithm is summarized in Figure 5.3.

```
1. get timing constraint $T$ , timing yield $\gamma$ and capacitive load $C_L$
2. while ( $N < \ln C_L$ )
       Solve (5.1)
       if ( $status$ = 'feasible' )
           goto step 3
       endif
       set $N = N + 1$ and goto step 2
3. adapt_buffer ( $N$ )

adapt_buffer ( $N$ )
    1. set $\alpha_l$ , $\alpha_u$ , $\Delta\alpha$
    2. set $\alpha = \alpha_l$ , $min = \infty$ , $trunc\_point = 0$
    3. while ( $\alpha \le \alpha_u$ )
           solve (5.14) for $l_0 = \mu_L + \phi^{-1}(\alpha)\sigma_L$
           if ( $obj\_value < min$ )
               $min = obj\_value$
               $trunc\_point = l_0$
           endif
```

Figure 5.3: Pseudo-code for solving the adaptive buffer design problem

5.3.1 Extension to Variability in Threshold Voltage

In the previous sections, we restricted attention to variability in L. We now show that the optimization strategy can be extended to two-dimensional uncertainty. Once again, we seek to find an optimal partition of the uncertainty set such that the design options corresponding to these partitions result in minimum total power consumption. Letting $\tilde{L}$ and $\tilde{V}_t$ denote the realizations of channel length and threshold voltage respectively, the buffer selected depends on the region where $\tilde{L}$ and $\tilde{V}_t$ lie – the first buffer is selected when $(\tilde{L}, \tilde{V}_t) \in [l_0, \infty) \times [v_0, \infty)$, and so forth. Denoting the probability of picking buffer j by α_j,

$$\alpha_1 = P((\tilde{L}, \tilde{V}_t) \in [l_0, \infty) \times [v_0, \infty)) = P(\tilde{L} \in [l_0, \infty))P(\tilde{V}_t \in [v_0, \infty)) \tag{5.15}$$

since ΔL and ΔV_t are independent random variables. The values of $\alpha_j, j \in [2, 4]$ can be similarly obtained.

The expected value of leakage of buffer j can be computed using Equation (5.13) . The results derived in Section 5.2, are valid for ΔV_t, thus the adaptable problem can be formulated for variability in L and V_t, similarly to (5.14). This is again a GP for a given truncation point (l_0, v_0). We obtain the optimal partition by a linear sweep in both l_0 and v_0.

5.4 EXPERIMENTAL RESULTS

The optimization problem was solved using the geometric optimization package *ggplab*, which is a toolbox implemented in MATLAB [72]. The experiments were run on a 32-bit, 3.7 GHz. Intel Xeon processor with 4GB of memory. The inverters were characterized using a 65nm process using Berkeley Predictive Technology Model [61]. An important part of the proposed strategy is the ability to measure the values of L and V_t. The modified shift-and-ratio method [73] and the technique based on the Drain Induced

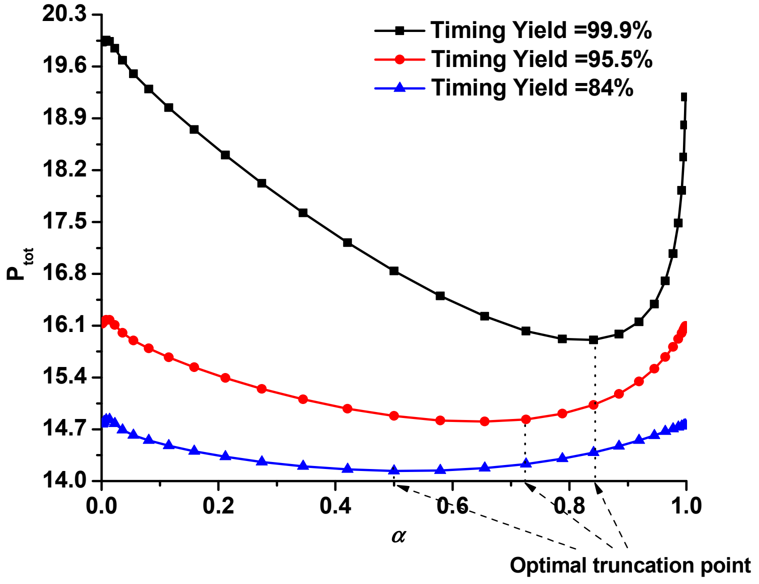


Figure 5.4: Dependence of total power of buffer configuration truncation point for different timing yields

Barrier Lowering (DIBL) phenomenon [74] have been demonstrated to be accurate for channel length extraction in sub-100 nm devices. For threshold voltage, several techniques have proven to be accurate [75].

Once the actual realizations of the uncertainty are known, the value of the appropriate buffer can be chosen based on the partition set that the uncertain variables lie in. Figure 5.3 shows the dependence of the total power of the configuration on the truncation point for different values of timing yield. It is clear that there exists an optimal partition for which the total power is minimized. The optimal partition at 99.9% timing yield is: proportion of low-power buffer use is 0.85 and high-speed buffer use is 0.15 Also, the optimal truncation point shifts towards the left as the yield constraint is relaxed. Picking the truncation point $\alpha \approx 1$, corresponds to solving (5.1) for the required timing yield constraints. As the figure shows, the adaptive scheme produces a savings of ~15%. The savings become greater as the magnitude of variability increases.

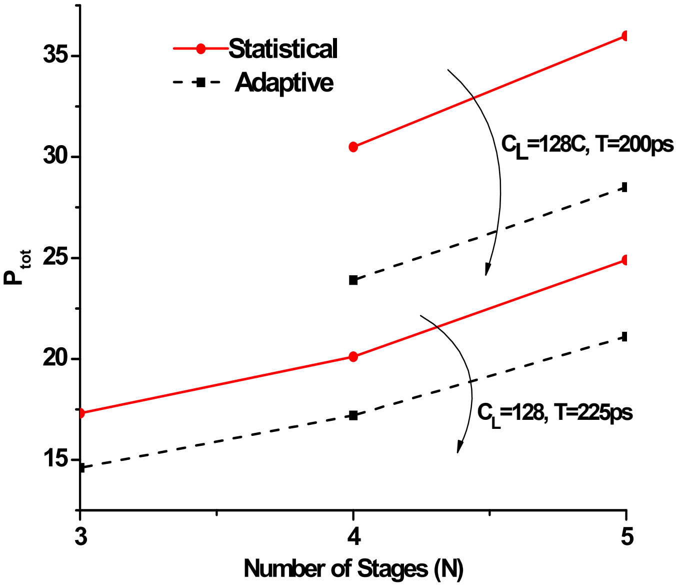


Figure 5.5: Comparison of statistical and adaptive buffer design approaches

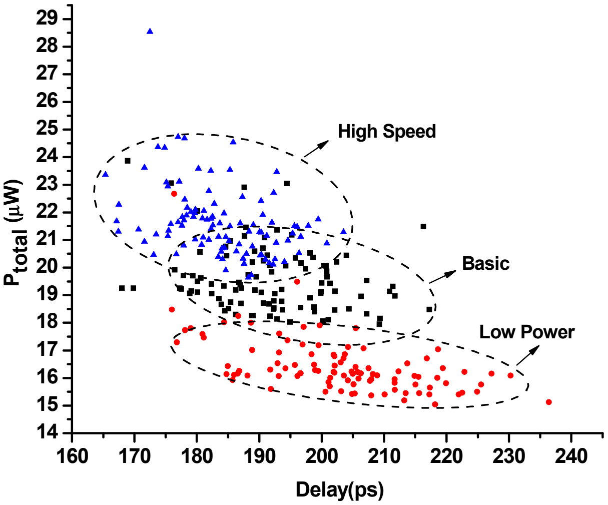


Figure 5.6: Monte Carlo simulation of buffer configuration

Figure 5.5 depicts the power- savings enabled by the adaptive scheme. The run-time behavior of the algorithm is very good. For instance, solving (5.14) to obtain an optimal truncation point and the corresponding buffer sizes takes ~15 seconds for a 5-

stage buffer configuration. Figure 5.6 shows a Monte-Carlo simulation of the buffer configuration. It is evident that the spread in power grows for faster configurations. This makes switching to the low-power version whenever possible a more attractive proposition.

5.5 SUMMARY

In this chapter we have developed an analytical post-synthesis and design-time co-optimization technique for buffer sizing using the paradigm of finite adaptable optimization. This approach provides the flexibility to pick one buffer from a finite number of buffer designs based on the realization of uncertainty. Results indicate that up to 15% savings in total power can be achieved by applying our technique, at the cost of 14% control logic area overhead.

Chapter 6: Conclusions

It is now universally accepted that deterministic timing and power optimization algorithms are no longer adequate as they do not handle variability effectively. The objective of this dissertation is to develop robust optimization algorithms that consider the impact of parameter variability on circuit timing and power consumption. This dissertation has investigated: (i) a gate sizing approach for area minimization under timing variability; (ii) an algorithm for total power minimization considering variability in timing and power (iii) a methodology for optimization of leakage power using design-time sizing and post silicon tuning using adaptive body bias; (iv) an optimization technique to minimize the total power of a buffer chain while considering the finite nature of adaptability afforded.

Timing yield is the percentage of chips meeting a specific constraint. Timing yield optimization strategies aim at ensuring that the circuit meets its timing constraints with a specific probability. Gate and transistor sizing that are often performed at the post-synthesis stage of design offers a simple strategy for yield optimization. It is natural to seek a statistical solution that can reuse the existing sizing tools and flows, and in this dissertation we develop an efficient statistical sizing algorithm for timing yield improvement. Specifically, variability in circuit delay is analytically treated by formulating a robust linear program with ellipsoidal uncertainty. This is then mapped onto a second-order conic program which can be solved efficiently.

However, in the nanometer regime, parametric timing yield alone is not a sufficient metric as it ignores variability in leakage power. This necessitates the development of optimization techniques to minimize parametric yield loss resulting from power and delay variability. In the absence of substantial leakage power, parametric yield is determined by the maximum possible clock frequency. Switching power is relatively insensitive to process variation. When the leakage power typical of current CMOS technologies is added, the total power starts approaching the power limit determined by the cooling and packaging considerations. Crucially, the exponential dependence of leakage on process spread means that the total power may cross the cooling (power) limit well below the maximum possible chip frequency, since chips operating at higher frequencies have exponentially higher leakage power consumption. Thus, due to the inverse correlation between speed and leakage, yield is limited both by slower chips and chips that are too fast, because they are too leaky. In this dissertation we propose an algorithm for total power minimization under timing and power yield constraints in the presence of variability. The algorithm is formulated as a robust optimization program with a guarantee of power and timing yields, with both power and timing metrics being treated probabilistically. Power reduction is performed by simultaneous sizing and dual threshold voltage assignment.

Parametric yield loss due to variability can be effectively reduced by both design-time optimization strategies and by adjusting circuit parameters to the realizations of variable parameters. The two levels of tuning operate within a single variability budget, and because their effectiveness depends on the magnitude and the spatial structure of variability their joint co-optimization is required. Algorithmically, future robust circuit synthesis can be conceptualized as a two-stage optimization problem, with additional second-stage tuning available upon the realization of uncertain variables. In this paper an efficient formulation is proposed using the theory of adjustable optimization. This optimization paradigm presumes that the decision-maker has a chance to update his optimization strategy upon learning additional information. We describe an optimization strategy that unifies design-time gate-level sizing and post-silicon adaptation using adaptive body bias at the chip level. In addition three measures of complexity that parameterize the solution and the optimality of this problem are introduced by us: the control complexity (the granularity of control), the measurement complexity (the granularity of the monitoring and sensing circuitry), and the parameter complexity (a measure of how spatially uncorrelated the process variable is). Using these metrics, formal quantitative trade-offs between design-time and post-silicon adaptivity can be identified.

Adaptable optimization seeks to make optimal design decisions in the first phase, given that such tuning is possible. Post-silicon tuning options are limited by physical constraints, as well as by the expense and implementation overhead of having to solve very complex post-silicon tuning optimization problems for every chip. Finite adaptable optimization is a framework intended for such limited second-stage adaptability. The second stage decisions are discrete, and chosen from a finite set of pre-determined "contingency" plans. Here we develop a strategy using finite adaptable optimization that enables reduction in power in the presence of variability by using a tunable buffer circuit: depending on realizations of process parameters, buffer stages with different size are selected and their thresholds are adjusted through body bias to minimize power while guaranteeing performance.

In this dissertation, we have analyzed the impact of variability on power and timing and its impact on circuit performance and yield and developed techniques to counter its detrimental effect. With high-end designs experiencing a double-sided squeeze on parametric yield due to the power-dissipation limits, power variability needs to be explicitly taken into account. This requires the adoption of new analysis and optimization methodologies and is expected that continued progress in this area will help designers deal with variability in a far more effective fashion.

Bibliography

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

[2] H. Chang, and S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal," *in Proc. International Conference on Computer Aided Design*, 2003, pp. 621 – 625.

[3] A. Devgan and C. Kashyap, "Block-based static timing analysis with uncertainty," *in Proc. International Conference on Computer Aided Design*, 2003, pp. 604-614.

[4] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst-case timing analysis*," in Proc. Design Automation Conference*, 2002, pp. 556–569.

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

[6] D. Boning and S. Nassif, "Models of Process Variations in Device and Interconnect," *Design of High-Performance Microprocessor Circuits*, A. Chandrakasan (ed.), 2000.

[7] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variation and impact on Circuits and Microarchitecture," in *Proc. of Design Automation Conference,* 2003, pp. 338-342.

[8] S. Nassif, "Delay Variability: Sources, Impact and Trends," in *Proc. of International Solid-State Circuits Conference*, 2000, pp. 368-369.

[9] V. Mehrotra, S. Nassif, D. Boning, and J. Chung, "Modeling the effects of manufacturing variation on high-speed microprocessor interconnect performance," in *International Electron Devices Meeting Technical Digest*, 1998, pp.767-770.

[10] D. Fitzgerald, "Analysis of polysilicon critical dimension variation for submicron CMOS processes," *M.S. thesis, Dept. Elect. Eng. Comp. Sci., Mass. Inst. Technol., Cambridge*, June 1994.

[11] M.D. Lenevson, N.S. Viswanathan, and R.A. Simpson, "Improving resolution in photolithography with a phase-shifting mask," *IEEE Transactions on Electron Devices*, vol. 29 (11), pp. 1828-1836, 1982.

[12] B. Stine, D.S. Boning, and J.E. Chung, "Analysis and decomposition of spatial variation in integrated circuit processes and devices," *IEEE Transactions On Semiconductor Manufacturing*, vol. 1, pp. 24-41, Feb. 1997.

[13] E. Chang, B. Stine, T. Maung, R. Divecha, D. Boning, J. Chung, K. Chang, G. Ray, D. Bradbury, O. S. Nakagawa, S. Oh, and D. Bartelink, “Using a Statistical Metrology Framework to Identify Systematic and Random Sources of Die- and Wafer-level ILD Thickness Variation in CMP Processes,” in *Proc. of International Electron Devices Meeting*, 1995, pp.499-502.

[14] B. Stine, D. Ouma, R. Divecha, D. Boning, and J. Chung, “A Closed-Form Analytic Model for ILD Thickness Variation in CMP Processes,” in *Proc. of CMP-MIC*, pp. 266. 273, 1997.

[15] D. Burnett, K. Erington, C. Subramanian, and K. Baker, “Implications of Fundamental Threshold Voltage Variations for High -Density SRAM and Logic circuits,” in *Proc.* Of *Symposium on VLSI Technology*, 1994, pp. 15-16.

[16] A. Keshavarzi, G. Schrom, S. Tang, S. Ma, K. Bowman, S. Tyagi, K. Zhang, T. Linton, N. Hakim, S. Duvall, J. Brews, and V. De, “Measurements and modeling of intrinsic fluctuations in MOSFET threshold voltage,” in *Proc. of International Symposium on Low Power Electronics and Design*, 2005, pp.26-29.

[17] K. Takeuchi, T. Tatsumi, and A. Furukawa, “Channel Engineering for the Reduction of Random-Dopant-Placement-Induced Threshold Voltage Fluctuations,” in *International Electron Devices Meeting Technical Digest,* 1997, pp. 841-844.

[18] K. Bowman and J. Meindl, “Impact of within-die parameter fluctuations on the future maximum clock frequency distribution,” in *Proc. of IEEE Custom Integrated Circuits Conference*, 2001, pp. 229-232.

[19] S. Nassif, “Within-chip variability analysis,” in *International Electron Devices Meeting Technical Digest*, 1998, pp. 283-286.

[20] N. Hakim, “Tutorial on Statistical Analysis and Optimization,” *Design Automation Conference,* 2005.

[21] S. Nassif, “Statistical worst-case analysis for integrated circuits,” *Statistical Approaches to VLSI*, Elsevier Science, 1994.

[22] M. Orshansky, J.C. Chen, and C. Hu, “A Statistical Performance Simulation Methodology for VLSI Circuits,” in *Proc. of Design Automation Conference,* 1998, pp. 402-407.

[23] Coudert, O., “Gate sizing: A general purpose optimization approach.” , *Proc. of EDTC'96*,, 1996.

[24] M. Borah, R.M. Owens, M.J. Irwin, “Transistor sizing for minimizing power consumption of CMOS circuits under delay constraint”, *Proc. of ISLPED*, 1995, pp. 167-172.

[25] Fishburn J., Dunlop A., “TILOS: A posynomial programming approach to transistor sizing”, *IEEE Trans. on CAD*, pp. 326-336.

[26] Berkelaar M., Jess J., “Gate sizing in MOS digital circuits with linear programming.”, *Proc. European Design Automation Conference*, 1990, pp. 217-221.

[27] Chen C. et al, “Fast and exact simultaneous gate and wire sizing by Lagrangian relaxation”, *IEEE Trans. on CAD*, 1999, pp. 1014-1025.

[28] Jacobs E. and Berkelaar M., “Gate sizing using a statistical delay model”, in *Proc. DATE*, 2000, pp. 283-290.

[29] Raj S. et al, “A methodology to Improve Timing Yield”, *Proc. of DAC*, 2004, pp. 448-453.

[30] Seung P. *et al*, “Novel sizing algorithm for yield improvement under process variation in nanometer technology”, *Proc. of DAC,* 2004, June 7-11, 2004, pp. 454 – 459.

[31] Patil D. *et al.*, “A New Method for Design of Robust Digital Circuits,” *Proc. of ISQED*, 2005, pp. 676-681.

[32] Singh, J. et. al., “Robust gate sizing by geometric programming”, *Proc. Of DAC,* 2005, pp. 315-320.

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

[34] R. Rao, A. Devgan, D. Blaauw, and D. Sylvester, “Parametric Yield Estimation Considering Leakage Variability,” in *Proc. of Design Automation Conference*, 2004, pp. 442-447.

[35] D. Lee, D. Blaauw, and D. Sylvester, “Gate Oxide Leakage Current Analysis and Reduction for VLSI Circuits,” *IEEE Transactions on Very large Scale Integration (VLSI) Systems*, vol. 12(2), pp.155-166, February 2004.

[36] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, “Parameter variation and impact on Circuits and Microarchitecture,” in *Proc. of Design Automation Conference,* 2003, pp. 338-342.

[37] D. Nguyen, A. Davare, M. Orshansky, D. Chinnery, B. Thompson, and K. Keutzer, “Minimization of dynamic and static power through joint assignment of threshold voltages and sizing optimization,” in *Proc. of International Symposium on Low Power Electronics and Design*, 2003, pp. 158 – 163.

[38] J. Tschanz, et al., “Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage,” *ISSCC Tech. Dig.*, pp. 422-423, 2002.

[39] Narendra S., Antoniadis D., and De V., “Impact of using adaptive body bias to compensate die-to-die *Vt* variation on within-die *Vt* variation”, *Proc. ISLPED*, pp. 229-232, 1999.

[40] Keshavarzi A., et al., “Effectiveness of reverse body bias for leakage control in scaled dual Vt CMOS ICs,” *Proc. ISLPED*, pp. 207-212, 2001.

[41] Martin S. et. al., “Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads,” *Proc. ICCAD*, pp. 721-725, 2002.

[42] Chen T. et al., “Comparison of Adaptive Body Bias (ABB) and Adaptive Supply Voltage (ASV) for improving delay and leakage under the presence of process variation*,” IEEE Trans. VLSI Sysems*, v. 11, no. 5, pp. 888-899, Oct. 2003.

[43] J. M. Rabaey, et. al., *Digital Integrated Circuits: A Design Perspective*, 2nd ed., Pearson Education Int., 2003.

[44] B. S. Amrutur and M. A. Horowitz, “Speed and power scaling of SRAM’s,” *IEEE J. Solid-State Circuits*, vol. 35, no. 2, pp. 175–185, Feb. 2000.

[45] S. Chen*, et. al*. “A new output buffer for 3.3-V PCI-X application in a 0.13-/spl mu/m 1/2.5-V CMOS process,” *Proc. Asia-Pacific Conference on Advanced System Integrated Circuits*, 2004, pp. 112-115.

[46] S. Tam, *et. al.*, “Clock generation and distribution for the first IA-64 microprocessor,” *Solid-State Circuits, IEEE Journal of* , vol.35, no.11, pp.1545-1552, Nov 2000.

[47] R. Ho, *et. al.,*“The future of wires,” *Proceedings of the IEEE* , vol.89, no.4, pp.490-504, Apr 2001.

[48] H. Kaul *et. al.*, “A novel buffer circuit for energy efficient signaling in dual-VDD systems,” *Proc of GLSVLSI*, 2005, pp. 462-467.

[49] G. Villar, *et. al.*, “Energy optimization of tapered buffers for CMOS on-chip switching power converters,” *Proc of ISCAS,* 2005, pp. 4453-4456.

[50] J. Xiong, *et. al*., “Buffer Insertion Considering Process Variation,” *Proc. of DATE*, 2005, pp. 970-975.

[51] V. Khandelwal, *et al.*, “A probabilistic approach to buffer insertion,” *Proc. of ICCAD,* 2003, pp. 560-567.

[52] H. Wang *et al.*, “Variable tapered pareto buffer design and implementation allowing run-time configuration for low-power embedded SRAMs,” *IEEE Trans. on VLSI*, Vol. 13, Oct. 2005.

[53] Boyd S., Vandenberghe L., *Convex Optimization*, Cambridge, 2004.

[54] Alizadeh F., Goldfarb D.,"Second-order cone programming", *Technical Report RRR*, Report number 51-2001, RUTCOR, Rutgers University.

[55] Y. Zhan, A. J. Strojwas, M. Sharma, and D. Newmark, “Statistical critical path analysis considering correlations,” *in Proc. International Conference on Computer Aided Design*, 2005, pp. 699- 704.

[56] X. Li, J. Le, M. Celik, and L. Pileggi, “Defining statistical sensitivity for timing optimization of logic circuits with large-scale process and environmental variations,” *in Proceedings of the 2005 International Conference on Computer-Aided Design*, 2005, pp. 844-851.

[57] http://www.mosek.com/documentation.html#manuals.

[58] Papoulis A., *Probability Random Variables and Stochastic Processes*, New York: McGraw-Hill, 1984.

[59] Ben-Tal, A. Nemirovski, A., “Robust Convex Optimization*,” Math. Oper. Res.,* 23, 1998.

[60] Ben-Tal A., Goryashko A., Guslitzer E., Nemirovski A., Adjustable robust solutions of uncertain linear programs, *Mathematical Programming*, Volume 99, Issue 2, Mar 2004, pp. 351 – 376.

[61] Cao Y. et al., “New paradigm of predictive MOSFET and interconnect modeling for early circuit design,” *Proc. of IEEE CICC*, 2000, pp. 201-204.

[62] S. Sirichotiyakul, T. Edwards, C. Oh, J. Zuo, A. Dharchoudhury, R. Panda, and D. Blaauw, “Stand-by power minimization through simultaneous threshold voltage selection and circuit sizing,” *in Proc. Design Automation Conference*, 1999, pp. 436-441.

[63] Q. Wang and S. Vrudhula, “Static power optimization of deep submicron CMOS circuit for dual V_{th} technology,” *in Proc. International Conference on Computer Aided Design*, 1998, pp. 490-496.

[64] M. Pelgrom, A.Duinmaijer, A. Welbers, “Matching Properties of MOS Transistors,” *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433-1440, Oct. 1989.

[65] S. Chakraborty and R. Murgai, “Complexity of Minimum-delay Gate Resizing,” *in Proc. International Conference on VLSI Design*, 2000, pp. 425-430.

[66] D. Markovic, V. Stojanovic, B.Nikolic, M. A. Horowitz, and R. W, Brodersen, “Methods for true energy-performance optimization,” *Journal of Solid-State Circuits*, vol. 39, no. 8, pp. 1282- 1293, Aug. 2004.

[67] V. Sundararajan, S.S. Sapatnekar, K. K. Parhi, “Fast and Exact Transistor sizing Based on Iterative Relaxation,” *IEEE Transactions on Computer Aided Design*, vol. 21, no. 5, pp.568-581, May 2002.

[68] C. Chatfield, *Introduction to Multivariate analysis*, Chapman and Hall, 1980.

[69] M. Romeo, V. Da Costa, and F. Bardou, “Broad distribution effects in sums of lognormal random variables,” *Eur. Phys. J.*, B 32, pp. 513-525, 2003.

[70] D. G. Chinnery, and Keutzer, K., “Linear Programming for Sizing, Vth and Vdd Assignment,” *in Proc. International Symposium on Low Power Electronics and Design*, 2005, pp. 149-154.

[71] C. Caramanis, *Adaptable Optimization: Theory and Algorithms*, PhD dissertation, Massachusetts Institute of Technology, June 2006.

[72] http://www.stanford.edu/~boyd/ggplab/

[73] C-W. Eng *et. al.,* “An Improved Shift-and-Ratio Effective Channel Length Extraction Method for Metal Oxide Silicon Transistors with Halo/Pocket Implants,” *Jpn. J. Appl. Phys.,* pp. 2621-2627

[74] Q. Ye and S. Biesemans, “Leff extraction for sub-100 nm MOSFET devices,” *Solid-State Electronics,* Volume 48, Issue 1, , January 2004, pp. 163-166.

[75] A. Ortiz-Conde, *et. al.*, “A review of recent MOSFET threshold voltage extraction methods,” *Microelectronics Reliability*, Volume 42, Issues 4-5, April-May 2002, pp. 583-596.

Vita

Murari Mani was born in Hyderabad, India in November, 1981. He received the B.E degree in electrical and Electronics Engineering from Madras University in 2003 and his M.S. degree from the University of Texas at Austin in 2005. His research interests include optimization techniques for yield improvement, statistical static timing analysis and design for manufacturability of digital integrated circuits. Murari Mani was the recipient of the Design Automation Conference Best Paper Award in 2005 and the IEEE William J McCalla Best Paper Award (International Conference on Computer-Aided Design 2006).

Permanent address: 'Anugraha', No. 702, 22nd Cross, 23rd Main, Ideal Homes
Township, Rajarajeshwari Nagar, Bangalore – 560098, India.

This dissertation was typed by the author.