3-23-2009

Techniques for VLSI Circuit Optimization Considering Process Variations

Mahalingam Venkataraman
University of South Florida

Techniques for VLSI Circuit Optimization Considering Process Variations

by

Mahalingam Venkataraman

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Nagarajan Ranganathan, Ph.D.
Srinivas Katkoori, Ph.D.
Hao Zheng, Ph.D.
Justin E. Harlow III, M.S.
Sanjukta Bhanja, Ph.D.
Kandethody M. Ramachandran, Ph.D.

Date of Approval:
March 23, 2009

Keywords: Variation Awareness, Circuit Design, Gate Sizing, Incremental Timing Placement, Buffer Insertion, Clock Stretching, Fuzzy Programming, Logic Level, Layout Level

DEDICATION

To my parents, with all my love and respect.

ACKNOWLEDGEMENTS

I would like to thank Dr. Nagarajan Ranganathan for giving me this opportunity to work with him. I am grateful to him for his enthusiasm for research, moral support and steadfast friendship. I cannot imagine how he comes up with the right questions to ask, whenever we have a research discussion on a new topic. His technical mentoring has made me a better researcher and his mentorship in teaching moral values has made me a better person. Inspite of his busy schedule, Dr. Ranga is always available as a friend to discuss personal issues as well. He is an ideal role model for aspiring students and I am lucky to have known and worked with him. I would also like to thank my committee members, Dr. Srinivas Katkoori, Dr. Hao Zheng, Prof. Justin E. Harlow III, Dr. Sanjukta Bhanja and Dr. Kandethody Ramachandran for the valuable time they took to review this thesis and their helpful comments. My sincere acknowledgments to Semiconductor Research Corporation (SRC grant 2007-HJ-1596) and National Science Foundation (NSF Computing Research Infrastructure grant CNS-0551621) for supporting this research in parts.

I am very grateful to my loving family without whose moral support this work would not have reached the completion. I cannot thank them enough for their constant source of inspiration. They have inculcated in me an attitude of working hard and being perseverant. I am indebted to them for being a perpetual source of inspiration and motivation for me. I cannot forget the help and support of all my friends and colleagues who have been with me in every step of this process. My friends inside and outside the research group have made the years spent in USF extremely pleasant. Several of their constructive criticisms and discussions have been useful in improving the quality of this research. Finally, thanks to Professor N. Venkateswaran (WARFT India) for identifying my research potential, getting me started and motivating me to pursue a Ph.D. degree.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Techniques for VLSI Circuit Optimization Considering Process Variations

Mahalingam Venkataraman

ABSTRACT

Technology scaling has increased the transistor's susceptibility to process variations in nanometer very large scale integrated (VLSI) circuits. The effects of such variations are having a huge impact on performance and hence the timing yield of the integrated circuits. The circuit optimization objectives namely power, area, and delay are highly correlated and conflicting in nature. The inception of variations in process parameters have made their relationship intricate and more difficult to optimize. Traditional deterministic methods ignoring variation effects negatively impacts timing yield. A pessimistic worst case consideration of variations, on the other hand, can lead to severe over design. In this context, there is a strong need for re-invention of circuit optimization methods with a statistical perspective. In this dissertation, we model and develop novel variation aware solutions for circuit optimization methods such as gate sizing, timing based placement and buffer insertion. The uncertainty due to process variations is modeled using interval valued fuzzy numbers and a fuzzy programming based optimization is proposed to improve circuit yield without significant over design. In addition to the statistical optimization methods, we have proposed a novel technique that dynamically detects and creates the slack needed to accommodate the delay due to variations.

The variation aware gate sizing technique is formulated as a fuzzy linear program and the uncertainty in delay due to process variations is modeled using fuzzy membership functions. The timing based placement technique, on the other hand, due to its quadratic dependence on wire length is modeled as nonlinear programming problem. The variations in timing based placement are modeled as fuzzy numbers in the fuzzy formulation and as chance constraints in the stochastic formulation. Further, we have proposed a piece-wise linear formulation for the variation aware buffer insertion and driver sizing (BIDS) problem. The BIDS problem is solved at the logic level, with look-up table based approximation of net lengths for early variation awareness. In the context of dynamic variation com-

pensation, a delay detection circuit is used to identify the uncertainty in critical path delay. The delay detection circuit controls the instance of data capture in critical path memory flops to avoid a timing failure in the presence of variations. In summary, the various formulation and solution techniques developed in this dissertation achieve significantly better optimization compared to related works in the literature. The proposed methods have been rigorously tested on medium and large sized benchmarks to establish the validity and efficacy of the solution techniques.

CHAPTER 1

INTRODUCTION

The rapid progress in technology is having a profound impact on the advancement of engineering. The technology evolution is enabling the development of new and improved electronic products, which in turn facilitates the progress of multiple engineering areas. The demand for high accuracy and performance in multi-functional phones, cameras and laptops is greater than ever before. In addition to the traditional improvement in performance upgrades, the consumers are interested in battery life, reliability and green computing. The crucial objectives in developing these products are, multi-functional features, high performance with low energy, portable and at the same time be cost efficient. A popular paradigm to achieve these objectives is to scale down the dimensions of the basic circuit elements. The transition to lower technology generations, for high performance and denser integration capabilities, is becoming complex due to increase in leakage energy and reliability concerns. Plus, the downward scaling of technology is also gradually reaching the limits of ballistic transportation [52].

1.1 Motivation

The above mentioned issues are impacting all fields of engineering. In the context of digital circuit optimization, the following issues need to be addressed. First, the demand for low power devices has increased with the growth of mobile devices for longer battery life and green computing. Green computing in this context, refers to the use of computing resources efficiently during its lifetime and promotes recyclability and biodegradability [27]. Secondly, due to nanometer integration levels, wiring density and aspect ratios of metal lines have increased the coupling capacitance between neighboring interconnects. In the nanometer era, the coupling capacitances in adjacent nets are strong enough to cause timing failures in circuits. Hence, the circuit optimization techniques in addition to optimizing performance have to consider the conflicting objectives of power and reliability as well. The optimization objectives are inter-related, conflicting and have become more intricate in the nanometer regime.

Optimization of performance is not sufficient and it can introduce a severe penalty in power and reliability, which in turn affects the performance as well. The multi metric optimization is no longer a recommendation option, but a necessity.

Variability in nanometer very-large-scale-integrated (VLSI) circuits has continued to increase with the decrease in feature size of transistors. The variations in process parameters increase the challenges of simultaneously optimizing power, performance and reliability. The main causes of the variations are either due to environmental effects like changes in power supply voltage and temperature or due to physical effects like changes in transistor width, channel length, oxide thickness and interconnect dimensions. The physical effects due to the imprecision in the fabrication process control leads to randomness in the number of do-pant atoms in transistors [22]. The uncertainty due to these process parameter variations deeply impact the timing and power characteristics of the circuits. Identically designed circuits can have a large difference in timing and power characteristics leading to loss in parametric yield of circuits [72].

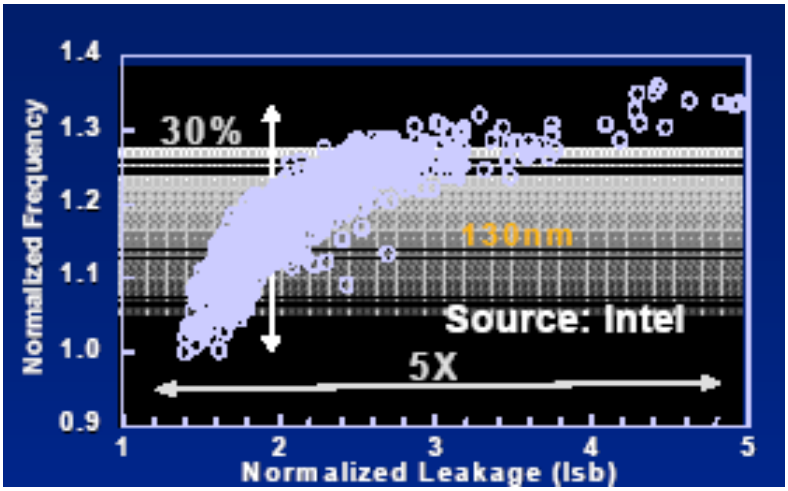


Figure 1.1 Impact of Process Variations on Power and Performance [72]

Figure 1.1 shows the deviation in leakage power and performance of Intel processors due to variations in process parameters. Hence, capturing the uncertainty due to these variations early in the analysis and optimization phase is crucial. Several circuit optimization methods over the years, have successfully improved area, power and timing of microprocessors. These optimizations make most paths of the circuits equally critical to achieve a optimal balance between power consumption and the timing specification. However, with the increasing effects of process variations in the nanometer era,

such optimization can worsen timing yield, as any of these critical paths can fail [95]. Here, timing yield is defined as the percentage of chips meeting the timing specification. A guarded approach to eliminate the effects of variability is to perform deterministic optimization at the worst case values of the varying parameters. The deterministic worst case approach guarantees high yield, but leads to high overhead in terms of circuit area, power and loss in optimization. On the other hand, the average case values for these parameters have less overheads, but may result in unacceptable timing yield. Hence, in the nanometer technology level, new methodologies are needed, which can guarantee high yield without compromising interms of area and power overheads.

The increasing effect of process variations in nanometer domain, has transitioned the design and optimization problem from the deterministic domain to the probabilistic domain [70, 72]. The process variations do not scale proportionally and their impact is increasing with each new technology node. In addition to increasing the complexity of finding an optimal solution, the circuit optimization process with a statistical perspective is inherently slower than their deterministic equivalent. In recent years, several research works have addressed variation awareness in the context of timing analysis and circuit optimization. Static timing analysis was replaced with statistical static timing analysis (SSTA) [11, 26, 74], where continuous distributions are propagated instead of deterministic values to estimate timing in presence of variations. More recently, statistical design optimization techiques [42, 49, 53, 54, 69, 95] for improving power and area for an acceptable yield have also been attempted. The optimization approach in [95], uses a penalty function to improve the slacks of critical paths to improve yield. An SSTA engine is used in the iterative optimization framework [49] to find the most critical gate to size in terms of power/delay sensitivity. In [53, 54], a stochastic programming approach with chance constraints is used to incorporate yield in the gate sizing problem formulation. However, the approaches using continuous distributions require a number of complicated operations to be performed iteratively at each node and hence incur a prohibitive runtime [20, 30]. The stochastic programming based statistical optimization technique [53, 54], on the other hand is fast, but more conservative [40] in terms of yield and hence lesser savings in terms of area and power consumption. Further, many of these methods are based on the assumption that the variation sources of the components follow specific distributions, such as Gaussian. However, manufacturing tests on fabricated circuits refute such assumptions and suggest against the use of assumptions on the distributions of variation parameters [44].

1.2 Contributions of Dissertation Research

In this dissertation, we propose the use of fuzzy mathematical programming for circuit optimization considering the uncertainty due to process variations. Fuzzy set theory can deal with multiple types of variations, specifically is popular in cases, where we cannot even predict the average behavior of the uncertain parameters. In fuzzy terminology, the above uncertainty is referred to as imprecision. Probability theory usually models situations where average behavior is predictable (situations that obey the law of large numbers) and enough information is available to model the probability distribution functions. The theory of fuzzy sets and systems, on the other hand, has been proven to successfully model imprecision and an interval value is sufficient to model variations. Fuzzy mathematical programming has successfully been applied in several areas of computer science and engineering like vision and robotics [64]. In VLSI circuit automation, fuzzy programming has also been used in VLSI testing and high level synthesis tasks like scheduling to model imprecise coefficients [30]. To the best of our knowledge, this is the first time the concepts of fuzzy sets and systems and fuzzy mathematical programming are being attempted at modeling uncertainty due to process variations in VLSI circuits.

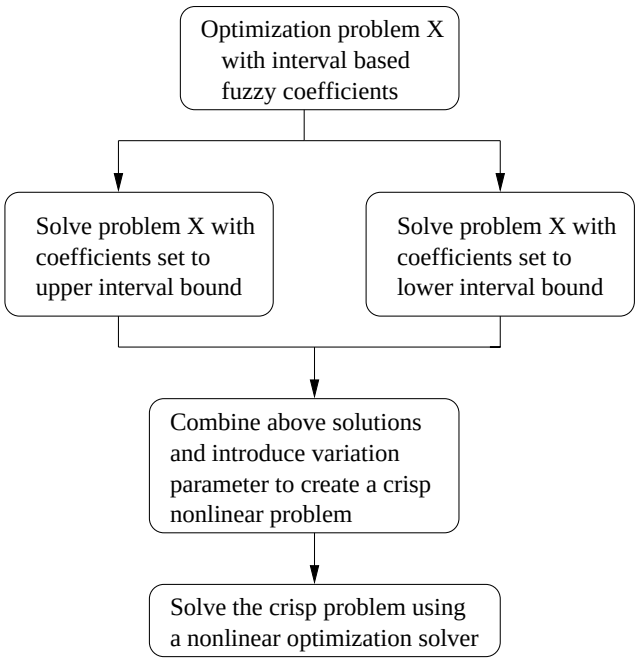


Figure 1.2 Fuzzy Linear Programming Approach for Variation Aware Optimization

Figure 1.2 shows an outline of the fuzzy optimization approach. Initially, a deterministic optimization is performed assuming the worst and the average case values for the variation parameters and

the results are used to convert the fuzzy optimization problem into a crisp nonlinear problem using the symmetric relaxation method [28, 64]. The crisp problem is then solved using a nonlinear optimization solver. The crisp problem in general, has been proved to provide the most satisfying solution in presence of imprecision or variations in coefficients of the constraints or objective function in the optimization problem [43, 56]. Next, we summarize the major contributions of this dissertation.

In this dissertation, we have proposed the use of fuzzy numbers for modeling uncertainty in delay due to process variations. Delays are modeled in an abstract fashion, considering the delay coefficients as an interval value with linear membership function. We have constructed fuzzy mathematical programming based formulations for gate sizing, timing based placement and buffer insertion to improve timing yield with minimal penalty in terms of design overheads. Fuzzy programming has been extensively used in civil, mechanical and computer science areas. However, this is the first time fuzzy techniques are employed for process variation aware VLSI circuit optimization. Secondly, we have also proposed a novel methodology for timing based placement using stochastic chance constrained programming (CCP). The CCP is formulated to capture the dependence of the constraints and objectives of the optimization using mean and variance of the uncertain parameters. We have compared the efficiency of fuzzy and stochastic CCP techniques for variation aware gate sizing and timing based placement. The results have confirmed the prediction in [40], that the efficiency of fuzzy mathematical programming is better than or comparable to stochastic CCP based optimization.

In the context of buffer insertion and driver sizing, we are the first to propose logic and layout level methodologies for simultaneous optimization of variation resistance, delay and power. The buffer insertion methodology was formulated as a variation aware piece-wise linear program at the circuit level. The circuit level methodology also overcomes the limitations of path based and net-based approaches. A look-up table based technique is used for predicting interconnect length at the logic level. The prediction technique is based on layout level simulations on sample benchmarks. The logic level approximation is shown to be comparable with detailed layout level results. Finally, we proposed a dynamic clock stretching technique for improving timing yield. Unlike the statistical design optimization methods, the clock stretching technique dynamically detects variations and incurs performance penalty only when there is a possible timing failure. The important contributions are summarized in Figure 1.3.

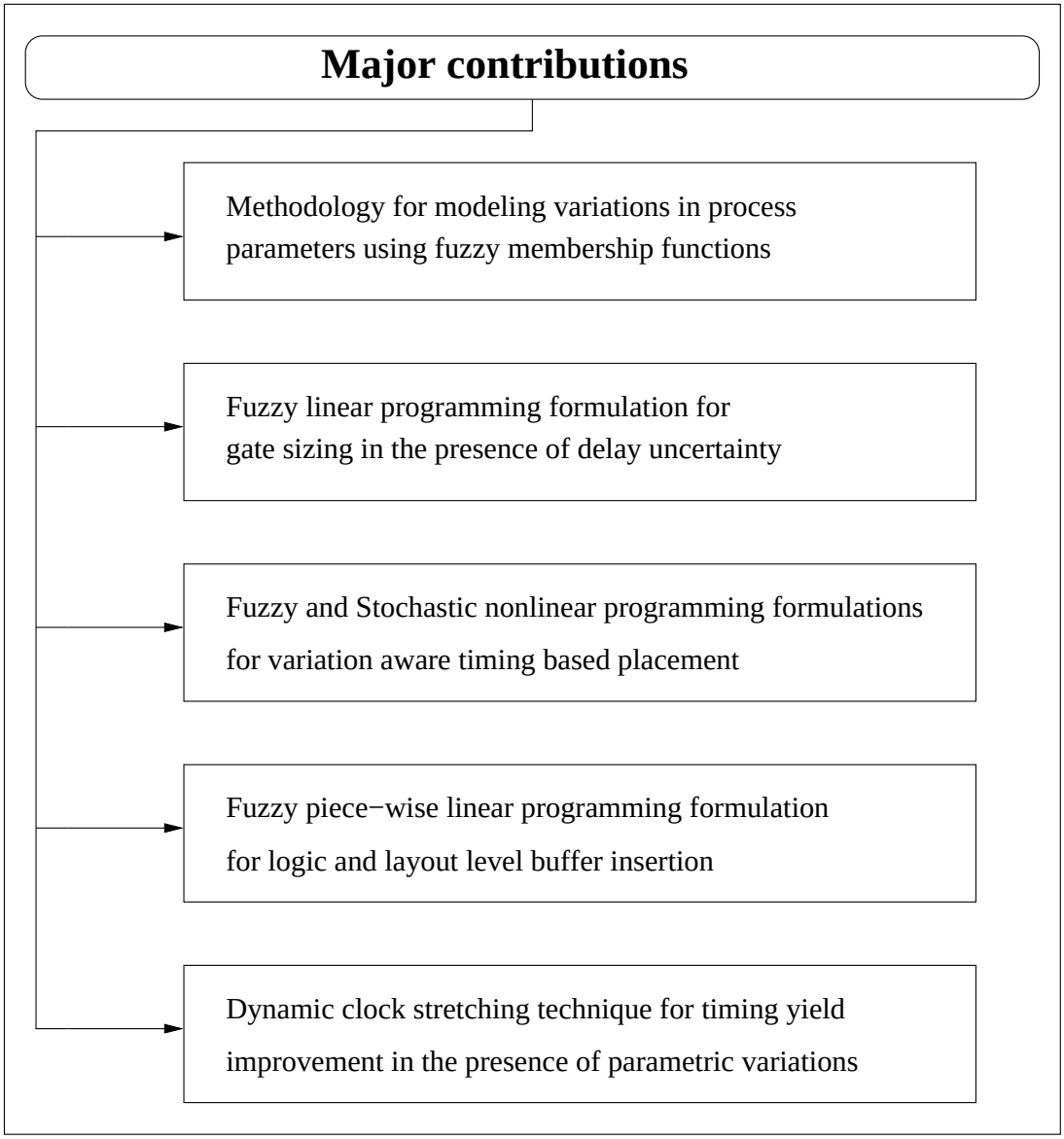


Figure 1.3 Major List of Contributions

1.3 Significance of Contributions

In this dissertation, we have proposed the use of fuzzy mathematical programming for uncertainty aware VLSI circuit optimization. The fuzzy optimization methodology is shown to abstractly model the variations in delay, with delay coefficients as interval values. Thus, the methodology can be used to model variations in process parameters even at logic level. Secondly, the fuzzy optimization methodology is shown to conveniently model variations in linear, nonlinear and piece-wise linear mathematical programming formulations. The above observation, infers that the fuzzy methodology is an effective

tool for several mathematical programming based VLSI circuit optimization problems. We have also shown that the fuzzy technique guarantees yield (evaluated using Monte-Carlo simulations) and effectively improves the objective function compared to the worst casing and stochastic chance constrained programming approach. Hence, the fuzzy mathematical programming (i) modeling, (ii) formulation and (iii) solution technique is a significant addition to the VLSI tools in the context of variation aware design.

A look-up table based interconnect length prediction technique is proposed and is used for logic level variation aware buffer insertion. The results of the logic level technique for variation aware buffer insertion is comparable to the values at the layout level, inspite of the approximation. The statistical design optimization using fuzzy programming is a design time technique to combat the effect of process variations on circuit performance. In this dissertation, we have also proposed a run time technique to dynamically detect delay due to variations and stretch the clock to avoid timing failures. The clock stretching methodology is an effective timing failure prevention technique and has less overhead compared to critical path isolation based clock stretching [76]. The dynamic technique does not require extra margin (over design) in the absence of variations. It is only activated in the presence of variations in delay on one of the top critical paths. Hence, the method stretches the clock only when necessary to avoid timing failure.

1.4 Dissertation Outline

The remainder of this dissertation is organized as follows. In Chapter 2, we describe the background that forms the basis of this research followed by relevant research contributions in areas related to the problems being addressed in this dissertation. The background discussion starts with the basic concepts of mathematical programming, importance of considering process variations in circuit optimization and briefly explains the fuzzy mathematical programming (FMP) and stochastic chance constrained programming (CCP) techniques for variation aware circuit optimization. Increasing levels of variations in process parameters are affecting performance and hence timing yield of VLSI circuits. In this context, we propose statistical and dynamic optimization techniques for improving timing yield without significant over design.

The fuzzy gate sizing (FGS) methodology for process variation aware optimization of power, delay and noise is presented in Chapter 3. The FGS technique is a post layout gate sizing approach, formulated as a linear programming problem with variations modeled as linear membership based fuzzy numbers. In Chapter 4, we describe both fuzzy and stochastic CCP based approach for timing based placement to optimize performance in the presence of process variations. The timing based placement problem is inherently nonlinear due to quadratic interconnect delay constraints and hence is modeled as a variation aware nonlinear programming problem. The process variations are modeled using fuzzy numbers with linear membership functions and chance constraints in FMP and CCP contexts respectively. We propose logic and layout level optimization techniques for variation aware buffer insertion and driver sizing in Chapter 5. The buffer insertion and driver sizing is formulated at the circuit level (instead of at the net or path level) and piece-wise linear constraints are used for modeling change in circuit delay. The variations in delay coefficients are modeled as fuzzy numbers with linear membership functions. In Chapter 6, we propose a dynamic technique for compensating the uncertainty in delay due to process variations. The methodology in the presence of variations uses a clock stretching logic circuit to increase the available slack and avoid a timing violation. The concluding remarks and future work in terms of extensions to the problems are addressed in Chapter 7 of this dissertation.

CHAPTER 2

BACKGROUND AND RELATED WORK

In this chapter, we present a brief introduction of the various concepts that form the basis for the research described in this dissertation. Specifically, we discuss the preliminaries of linear, non-linear programming, circuit optimization with multiple objectives, fuzzy mathematical programming technique and the stochastic chance constrained programming. The uncertainty based optimization techniques namely fuzzy and stochastic approaches are used for solving gate sizing, placement and buffer insertion considering process variations.

2.1 Design for Manufacturing

The shrinking design nodes, expanding design complexity and density in the nanometer dimensions are resulting in huge yield losses due to electro migration, leakage and integrity issues. Further, variations in process parameters at smaller geometries are adding to the design issues requiring additional rules to be administered before pattern generation. Physical fix-ups by processing mask data happen too late and cannot catch yield reducing problems. Hence, design engineers now need to address manufacturing issues throughout the design chain. Yield-savvy design tools and methodologies are required early in the design flow that can address performance, power, design functionality and manufacturing yield.

In the early nineties, the chip designers considered testing as an after concern. Even with increased number of test patterns and huge testing time, design engineers did not consider testability as a part of their design process. The test engineers in the manufacturing division was responsible for making the chip testable and have good fault coverage. Variation aware design for manufacturing is still considered as an after thought by some chip designers. Maximizing yield early in the design flow is only a recommended strategy and meeting physical and electrical design rules is sufficient. Yield optimizations and tweaks are still handled at the manufacturing environment. However, design optimization

without considering variations can worsen yield to irrecoverable levels at the manufacturing stage. In the 90 nm era, only slightly more than 40% of chip designs operate as expected and the rest need a complete mask re-spin to achieve acceptable performance and yield. Yield optimized standard cells, recommended design rules on via placement, statistical timing analysis and statistical optimization during sizing, buffer insertion are some popular design level techniques considering yield [57, 92]. Yield enhancing methodologies are highly interdependent on other metrics like power and performance as well and hence needs to simultaneously consider multiple metrics. Next, we introduce the various objectives and briefly give an overview of yield-aware statistical design techniques.

2.1.1 Optimization with Multiple Parameters

In the nanometer era, the term high performance of a VLSI circuit is not a simple function of delay or frequency of the circuit. The components such as power, crosstalk noise and process variations also have a large effect on delay and hence it is crucial to be considered as optimization metrics. A simultaneous optimization of these metrics (cost functions) is essential to design robust and reliable high performance circuits. Physical design steps like floor planning, standard cell placement and routing typically concentrate on area, wire length and delay. The computational complexity of these algorithms is huge for large sized benchmarks. Hence, the responsibilities of the circuit optimization techniques are not only maximizing performance, but also minimize power, crosstalk noise and effects due to variations in process parameters. The problem of circuit optimization primarily involves incremental changes or tuning of circuit components like gate sizes, threshold voltage, wire size, cell locations, and buffer insertion to maximize or minimize the overall objective or function [21]. Since, the focus of this dissertation is simultaneous optimization of power, delay, crosstalk noise and yield, the methods that are effective for the optimization for these metrics are discussed. However, the optimization framework developed in this dissertation is built from a generalized point of view and hence any existing or new metrics can be added or removed with minimal effort.

The transistor sizing, gate sizing and wire sizing problems are important in VLSI design because they enable us to explore the trade-offs between the multiple objectives in the cost function. The methodology (TILOS) in [2], use a convex programming based iterative transistor sizing technique based on critical delay sensitivity to improve performance. TILOS uses convex delay models for tran-

sistor sizing as they have the advantage that a local optimum is a global one. The iterative methodology was further improved in [80]. However, the general technique is not efficient for problems with more than a few thousand sizeable components. The authors in [1], proposed a lagrangian relaxation based gate sizing technique with simple constraints, that can be solved efficiently. The methodology however requires a good initial solution and sub gradient optimization step to converge practically. In a bid to improve computational complexity without significant impact on solution quality, the authors in [17], have proposed a robust linear programming framework for gate sizing. Secondly, in the context of increasing wire delay, buffer insertion and wire sizing are two popular methods to improve circuit performance. The delay of a net is directly proportional to the product of the resistance and capacitance along the wire. Since, they both are internally a function of the length of the net, the delay exhibits a quadratic dependence on the wire length. The objective of the buffer insertion problem is to divide the wire into a number of segments, such that the sum of delay of each wire segment becomes a linear function of the length of the total wire [68]. Van Ginneken in [47], presented a dynamic programming based optimal buffer insertion algorithm. The approach has quadratic complexity in the number of buffer locations and has been a foundation for several later works in the context of net-based buffer insertion.

The leakage current of a transistor can be controlled by fitting the circuit with a higher threshold voltage (Vth). The exponential dependence of leakage power on threshold voltage reduces the power consumption, with a delay penalty. Hence, several works [46, 61] have increased the Vth of the non-critical transistors to improve leakage power. The threshold voltage assignment step is usually combined with the gate/wire sizing methodology to efficiently optimize dynamic and leakage power. Timing driven placement is another important step in the physical design of integrated circuits. It can be formally defined as the process of finding the optimal locations of cells in a critical sub circuit such that the delay of the circuit is minimized. In timing based placement, the objective is to minimize the length of the critical interconnects, by incrementally adjusting the locations of cells [83, 91]. The timing of a circuit in the context of timing based placement is usually measured in terms of the worst negative slack and the total negative slack. The term slack in this context, is defined as the difference between the required time and actual arrival time of the signal. The incremental placement process creates overlaps, which is usually removed by a legalization step. The legalization step can impact timing in the reverse direction. However, with movement restrictions on cells over a localized neigh-

borhood, it was shown in [50], the reverse impact on timing is negligible. In the next section, we discuss in detail the importance of process variations and various techniques which are proposed to reduce the impact of variations on timing yield.

2.2 Process Variations

Process variations, in general, refer to the difference between the intended and obtained parameter dimensions prior and post fabrication of the circuit. With rapid scaling of technology into nanometer dimensions, the variations in semiconductor parameters like device length (L_{eff}), threshold voltage (V_{th}), gate oxide thickness is becoming fatal to circuit yield. The variations in process parameters can be classified as inter-die and intra-die variations. As the name suggests, inter-die variations are constant within a die but vary from one die to another die. The effect of inter-die variations can be detected during production test and a voltage or frequency change can prevent the loss of yield. Intra-die variations on the other hand, are variations that within a single die, meaning that a device/net parameter vary differently between different locations on the same die. Intra-die variations affecting doping concentration mainly result from fabrication equipment limitations. Intra-die variation also exhibits spatial correlation. Devices that are close together in the layout have a higher probability of being alike in characteristics than devices placed far apart. CMP (chemical-mechanical polishing) effects and optical proximity effects also increase the magnitude of intra-die variation in nanometer technology [14,89]. Hence, the amount of randomness in between gate/net within a die due to intra-die variations is large. Since, intra-die variations are random across the die, adjusting voltage or frequency to meet the worst case setting of all the parameters is often pessimistic. Hence, to meet the conflicting objectives of high performance, low power and high yield it is necessary to bring variation awareness in the design flow.

2.2.1 Delay Variations

The impact of intra-die process variations on delay is large as many factors can affect it. The delay for a gate and interconnect can be represented as,

$$gatedel_i = a_i - b_i s_i + c_i C_{load-j} \tag{2.1}$$

Equation 2.1 models gate delay as a linear function of gate size and load capacitance [17]. The gate delay can also be modeled using complex functions. However, from the context of gate sizing the model optimally trade-offs solution quality and execution time. The wire delay (Equation 2.2) can be represented as a quadratic function of interconnect length [83].

$$netdel_i = R_0 * len_i(0.5 * C_0 * len_i + C_{pin}) \tag{2.2}$$

The complete notations for these equations are described in later sections. The changes in effective length, oxide thickness and related process parameters can have significant impact on the coefficients of the above equations. It has been predicted in [12,77], that a delay variation of around 25% is possible in nanometer technology generations. In this work, we abstractly model the effects of variations on delay using the coefficients b_i, c_i, R_0 and C_0. Note that in addition to the process parameter variations, environmental factors such as power supply and temperature can also affect delay. However, we only consider the physical process parameter variation impacts as it is the dominating factor affecting the delays [77]. Plus, the above methodology can incorporate the impact of voltage and temperature variations by using more sophisticated models.

2.2.2 Statistical Optimization

Considering the sensitivity of circuit delay to process variations, several previous attempts focused on timing analysis. New approaches for both deterministic static timing analysis (STA) and statistical STA have been proposed. Previously in deterministic STA [2, 63], process variations have been modeled using case analysis. Thge best case, nominal and worst case parameter sets are constructed and timing analysis is repeated for each corner. The deterministic STA has a linear run time complexity as a function of the circuit size. However, in the nanometer era, with multiple variation sources, using the worst case parameters for intra-die variations leads to a pessimistic estimate. Hence, numerous works have proposed to replace STA with statistical STA [11, 67, 74]. The SSTA approaches however have high run time complexity due to re-converging paths in the circuit and complex computations.

Recently, several researchers have attempted to optimize power, delay and noise in the presence of process variations [10, 26, 49, 53, 54, 84, 87, 95]. The works in [10, 49] mainly focused on circuit optimization schemes with a statistical perspective. In other words, a statistical delay model or SSTA

is used to guide timing analysis. The authors in [53, 54], presented a statistical optimization approach that takes into account randomness in gate delays by formulating an efficient mathematical program. The major part of the work in this dissertation focuses on statistical optimization of yield, delay and power as a variation aware mathematical program formulation.

2.2.3 Dynamic Techniques for Variation Compensation

The increasing impact of variations in the nanometer era is necessitating the use of dynamic and run-time technique to combat to improve yield. Conventional techniques like scaling up supply voltage of upsizing logic gates over consume resources in the presence or absence of variations. The objective of these works has been to improve the power/overheads compared to worst case design while maintaining the timing yield. On the other end of the spectrum, design techniques have been proposed based on adaptive body biasing for post silicon process compensation [71]. Due to the quadratic dependence of supply voltage to dynamic power, researchers have proposed adaptive voltage scaling technique that are robust with respect to process variations and has limited over design. One such technique called Razor [24], uses dynamic detection of delay due to variations and corrects timing failures by adjusting the voltage of the processor. The Razor technique eliminates the need for voltage margins and hence can achieve significant savings in overheads compared to worst casing and statistical optimization. Secondly, the authors in [76], proposed a novel design paradigm which achieves robustness with respect to timing failure by using the concept of critical path isolation. The methodology isolates critical paths by making them predictable and rare under parametric variations. The top critical paths, which can fail in single cycle operation, are predicted ahead of time and are avoided by providing two cycle operations. The approach is extremely useful for certain design with rare critical paths (ex: adder). The critical path approach when tested on random designs, save power with a small timing penalty. Since, several circuit optimization techniques like, gate sizing, buffer insertion and incremental placement are inherently suited to be modeled as a mathematical program. In the next section, we discuss briefly the basics of mathematical programming and uncertainty aware optimization schemes.

2.3 Mathematical Programming

A mathematical program is either a maximization or minimization program to optimize an objective function with a set of constraints. In linear programming, the objective function and the constraints are always linear functions of the decision variables. A simple example of an linear programming problem can be shown as,

$$
\begin{aligned}
&maximize && c_1x_1 + c_2x_2 + ... + c_nx_n \\
&subjectto && a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \leq b_1 \\
& && a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \leq b_2 \\
& && . \\
& && . \\
& && a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \leq b_n \\
& && x_1, x_2, ...x_n \geq 0
\end{aligned}
\tag{2.3}
$$

Here, x_i is the decision variable, m is the number of constraints and n is the number of variables. The objective here is to maximize the objective function $c_1x_1 + c_2x_2 + ... + c_nx_n$. A feasible solution in linear programming context is the one which satisfies all the constraints and maximizes the objective function. Simplex method has been a simple, fast and efficient approach to solve linear programming problems. It is an iterative process in which, we start with a simple solution and then improve the same in a relative fashion. The iteration continues until no further improvement is possible.

The second class of mathematical optimization programs are broadly termed as the nonlinear programming (NLP). The main difference between the linear and the nonlinear problems is that the constraints and the objective functions are allowed to be nonlinear functions of the decision variables. A major challenge in NLP problems is the existence of local optima. Local optima is formally defined as spurious solutions that barely satisfies the requirements on the derivatives of the functions. Nonlinear optimization algorithms that overcome this difficulty are called global optimization algorithms. The nonlinear problems are further classified as convex and non-convex problems. Convex methods have a special property, where any local optimal solution is also a global optimum. Least squares, linear pro-

gramming, conic program, geometric program, quadratic program and semi definite programs are all examples of convex optimization programs [73]. The theory of convex sets and convex optimization is not new to VLSI and has been used in transistor sizing [2,35,80]. The delay of a gate in these works is modeled as a posynomial function of its transistor size using the Elmore delay model. A popular optimization package to solve convex programming problem is the MOSEK solver. It is defined to solve large scale linear, conic, quadratic, convex nonlinear and mixed integer problems.

Further, in addition to the optimization solvers, the representation of the optimization problem is also a crucial task. AMPL (A mathematical programming language) [65] and GAMS (General algebraic modeling system) are popular high-level modeling system for mathematical programming based optimization. A primary advantage of AMPL is the similarity of its syntax to the mathematical notation of optimization problems, which makes the optimization problems more readable. The optimization problem represented in a mathematical programming language can be solved using commercial optimization solvers. Several of these commercial software are also available for free usage at the NEOS server for optimization. The optimization problems can be submitted via email or on its website [66].

The coefficients of the mathematical programming problem are generally assumed to be exactly known. However, the assumption is not true for majority of the real world problems. Usually, the coefficients of the programming problem is either subject to errors of measurement or vary with technology or market conditions. Hence, it is crucial to introduce imprecision or uncertainty into the modeling process, while solving real world problems. A classical approach has been to use the concepts of probability theory. The probabilistic analysis is only proper for situations, which are reproducible and has happened a sufficient number of times. Therefore, the probabilistic techniques require some statistical data to provide information on the variation distribution of the random variable occurring in the mathematical model. The modeling of imprecision or uncertainty can also be done using fuzzy set theory and the concepts of fuzzy programming. The fuzzy programming techniques assume that the distribution of random variables are not exactly known and only belong to non-void sets or intervals. In the next sections, we discuss the preliminaries of variation aware optimization using fuzzy and stochastic programming.

2.4 Fuzzy Linear Programming Methodology

In this section, we briefly discuss important concepts in uncertainty aware optimization using fuzzy programming and variation modeling using fuzzy numbers. The reader is referred to [Sakawa 2002 [56]; Klir and Yuan 1995 [43]; and Zadeh 1970 [64]] for a detailed treatment of fuzzy mathematical programming. Zadeh (1965) introduced the concept of fuzzy sets and systems in which an element belonging to a set need not be binary valued, but could be any value in between [0, 1]. The membership value of the element in the interval [0, 1] is decided on how much it belongs to it and higher the degree of belonging, then, higher the membership value is.

2.4.1 Fuzzy Numbers

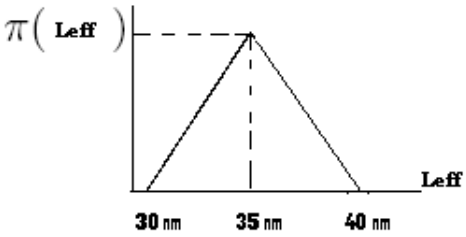
Fuzzy set theory and fuzzy optimization techniques provide an efficient mechanism for modeling and optimizing systems that exhibit imprecision. The theory and methodology of fuzzy programming based optimization has been popular since the inception of decision making in fuzzy environments by Bellman and Zadeh, in 1970 [64]. Several models and approaches have been proposed for uncertainty management using fuzzy linear programming, fuzzy multi-objective programming, fuzzy dynamic programming, fuzzy integer programming, possibilistic programming and fuzzy nonlinear programming. An extensive list of references can be found in [56] and [39]. A recent survey on fuzzy linear programming based optimization from a practical perspective has been provided by Inuiguchi and Ramik [51]. Hence, in order to solve a optimization problem with uncertain variable using fuzzy optimization, the principal requirement is an efficient fuzzy modeling of uncertainty.

A simple information such as ”the processing time for a task is around 23 minutes or within the range of 21 and 25” can be expressed by means of the following membership function,

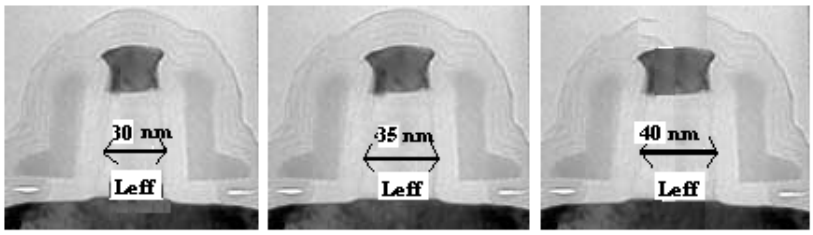
$$\mu_{23}(x) = \begin{cases} (x-21)/2 & \text{if } 21 \leq x \leq 23 \\ (25-x)/2 & \text{if } 23 < x \leq 25 \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

The uncertainty due to process variations are usually modeled as normally distributed random variables with mean E and standard deviation σ. In this work, instead of using normal distribution, we model these variations as interval valued fuzzy numbers in the range E-3σ and E+3σ. The 3σ, value

is assumed to be the deterministic worst case variation value, meaning all uncertain process parameters are set to 3σ, for maximum timing yield in worst case deterministic optimization. Interval valued fuzzy numbers were first explained by Zadeh, in [64], using possibilistic distributions. In the context of fuzzy mathematical programming, possibilistic distributions $\pi(.)$ are analogous to linear membership functions [30]. Figure 2.1 shows a symmetric triangular fuzzy number for modeling the variations in channel length of nanometer level transistors. Triangular and trapezoidal memberships are commonly used possibilistic distributions in solving fuzzy mathematical programming problems. Fuzzy optimization with nonlinear membership functions have also been attempted for improving modeling accuracy [56] .



Legend: Triangular fuzzy number modeling variations in channel length of 65 nm transistors.



Examples of transistors with and without channel length variations

Figure 2.1 Triangular Fuzzy Number Modeling for Varying Gate Length

The triangular fuzzy number in Figure 2.1 is commonly denoted by a triple X = (x^m, x^l, x^u), where x^m is the most possible value or the mean value and x^l, x^u are the lower and upper bounds, denoting the pessimistic and optimistic value of the number. Depending on the context, the value x^l can be pessimistic or optimistic variation from the mean value x^m and the same holds for the value x^u. In the context of VLSI circuit optimization, the triple $L_{eff} = (L^m_{eff}, L^l_{eff}, L^u_{eff})$ can be used to model variations in channel length. Since the general objective in circuit optimization is to minimize delay or power, the pessimistic value in this context for effective channel length is L^u_{eff}, which is the sum

of $L_{eff}^{m} + 3\sigma$. Similarly if we model the gate oxide thickness as a fuzzy triple, the pessimistic value is once again the upper bound value.

2.4.2 Solution Technique: Fuzzy Linear Programming

In this subsection, we explain the solution methodology of variation aware optimization using fuzzy linear programming. Fuzzy linear programming (FLP) is a special case of fuzzy mathematical programming, where objective function and constraints of the optimization problem are linear. The varying coefficients are assumed to vary linearly in the specified interval. The FLP problem shown here is a maximization problem with uncertain coefficient in the constraints.

$$\begin{aligned} &maximize \sum_{i=1}^{n} a_i x_i \\ &subject\ to \sum_{i=1}^{n} \tilde{b}_{ji} x_i \leq c_j, \quad 1 \leq j \leq m \end{aligned} \tag{2.5}$$

where, m is the number of constraints, n the number of variables and at least one $x_j > 0$. In the above optimization problem, the coefficient $\tilde{b}_{ji}$ is the interval valued fuzzy number which has a mean value, b_{ji} and a maximum variation of d_{ji}. The upper bound is assumed to be the pessimistic variation for this fuzzy number. The fuzzy number $\tilde{b}_{ji}$ is also assumed to vary linearly with the value of the variable x_i. To defuzzify the FLP we need to identify the lower and upper bounds of the optimal solution. The upper bound value for the fuzzy optimization problem can be estimated by setting the fuzzy coefficients fixed to the average case of the triangular fuzzy number as shown in the following equation.

$$\begin{aligned} Obj_1 &= maximize \sum_{i=1}^{n} a_i x_i \\ &subject\ to \sum_{i=1}^{n} b_{ji} x_i \leq c_j, \quad 1 \leq j \leq m \end{aligned} \tag{2.6}$$

Similarly, the lower bound value is found by setting the fuzzy coefficients to the pessimistic (assumed to be upper bound here) value,

$$Obj_2 = maximize \sum_{i=1}^{n} a_i x_i \tag{2.7}$$
$$subject\ to \sum_{i=1}^{n} (b_{ji} + d_{ji}) x_i \leq c_j, \quad 1 \leq j \leq m$$

Now, with these bound objective values and a new variation parameter, we can formulate a crisp problem, which will represent a optimal solution in presence of variations. The objective function for the fuzzy programming problem takes values between this lower Obj_l = min(Obj_1, Obj_2) and upper Obj_u = max(Obj_1, Obj_2) bound values. Using these bound values and the symmetric definition of fuzzy decision proposed by Bellman and Zadeh [64], the fuzzy problem can be de-fuzzified into a crisp nonlinear problem as shown in the following equation.

$$maximize\ \ \lambda \tag{2.8}$$
$$\lambda(Obj_l - Obj_u) - \sum_{i=1}^{n} a_i x_i + Obj_u \leq 0,$$
$$\sum_{i=1}^{n} (b_{ji} + \lambda d_{ji}) x_i - c_j \leq 0, \qquad 1 \leq j \leq m$$
$$x_j \geq 0,\ \ 0 \leq \lambda \leq 1 \qquad 1 \leq i \leq n$$

where, λ is the variation parameter introduced in the crisp problem which is to be maximized for an fuzzy optimal solution between the lower and upper bound values. The solution of this nonlinear programming problem can be interpreted as representing an overall degree of satisfaction in presence of varying parameters [56]. In the general formulation the variation parameter λ can take values between 0 and 1. In the next section, we explain the basic of our second technique, namely the stochastic chance constrained programming problem.

2.5 Stochastic Chance Constrained Programming

Programming under probabilistic constraints as a decision model under uncertainty, has been introduced by charnes, cooper and symonds [8]. The methodology was coined as chance constrained programming by these authors for this model, its extension and its variants [7]. Similar to the fuzzy programming technique, the chance constrained programming handles uncertainty in mathematical

programming based formulations. The chance constrained technique uses a relaxation step to convert the uncertain optimization problem into a crisp nonlinear programming problem. The focus of stochastic chance constrained programming is to increase feasibility of the program's ability to meet constraints in an uncertain environment. Next, we explain the solution technique of the chance constraint programming methodology. The discussion starts with the underlying linear programming problem, which can be shown as,

$$\begin{aligned} &minimize\, C^t x \\ &s.t.\ \ Ax \geq b,\ \ x \geq 0 \end{aligned} \tag{2.9}$$

Here, h and x are n-vectors, b is a m-vector and A is a m x n matrix. In several engineering problems, the estimation of the constraint matrix A is difficult and has a certain amount of uncertainty associated with it. In such situations, the system is required to satisfy the corresponding constraint with a probability $p = (0,1)$. The stochastic equivalent of the linear programming problem can be shown as,

$$\begin{aligned} &minimize\, c^t x \\ &s.t.\ \ P(Ax \geq b)\ \ \geq p,\ \ x \geq 0 \end{aligned} \tag{2.10}$$

The probability p, in the context of engineering type problems, may reflect the reliability of the system. The probability p ensures that the state of the system remains within a subset of all possible states. The subset mainly focuses on functioning of the system without major failures. The choice of the probability p is often arbitrary and accounts for the loss whenever constraints are violated. The above shown formulation can be efficiently transformed under the assumptions of node delay independence and Gaussian structured random values. The transformed probabilistic constraint can be expressed as follows,

$$\mu(Ax) + \phi^{-1}(1-p)(z^T C z)^{0.5} \geq b \tag{2.11}$$

Here, the assumption is that the random variable has a joint normal distribution with expectation μ and covariance C (matrix) and z is the standard deviation. The value of $\phi^{-1}(1-p)$ for the constraints can

be used to control the subset size and the associated reliability of the system. A high value for the above coefficient increases the yield and reliability of the system. Similar to the fuzzy programming solution, the stochastic chance constrained programming can also be solved using a nonlinear optimization solver. In this dissertation, we propose statistical design optimization solutions using both fuzzy and stochastic chance constrained programming for variation awareness in circuit optimization.

CHAPTER 3

VARIATION AWARE GATE SIZING

3.1 Problem Definition

Gate sizing is a simple yet powerful technique to improve the power/delay ratio of VLSI circuits. Several modeling schemes and solution methodologies have been proposed over the years to optimize power and performance through gate sizing. The process of gate sizing can be defined as finding the optimal drive strengths of individual gates of a circuit for a given objective function and constraints. For example, the objective function can be to minimize power or area for a specified timing target. Taxonomy of various gate sizing approaches, found in the literature, categorized as (i) deterministic [1,2,13,18,41,60,80,90] and (ii) variation aware gate sizing [10,15,26,35,42,49,53–55,58,59,69,78, 87,88,95] approaches is shown in Figure 3.1. The works listed in the Figure have used iterative sizing, linear programming, geometric programming, game theory and several other interesting formulations to identify the optimal gate sizes for minimizing the power and/or timing of circuits. The variation aware gate sizing works models the process, voltage and temperature (PVT) variation using a statistical approach. The PVT variations in the nanometer era, can be categorized as inter-die and intra-die variations. The inter-die variation occurs across different dies and affects all the transistors in the chip in a similar fashion. The intra-die variations, on the other hand, refer to variability within a single chip resulting in the gate lengths of some transistors larger and some others smaller than the intended sizes. The characteristics of the intra-die variations are correlated with respect to the position of the transistor in the die.

The modeling of process variations, initially was limited to statistical static timing analysis (SSTA) [11,26,74], where continuous distributions are propagated instead of deterministic values to find closed form expressions for performance in presence of variations. More recently, statistical design optimization for improving power and area for an acceptable yield has been investigated in [42,49,53,54,69,95]. In [95], the optimization uses a penalty function to improve the slacks of critical paths to improve yield.

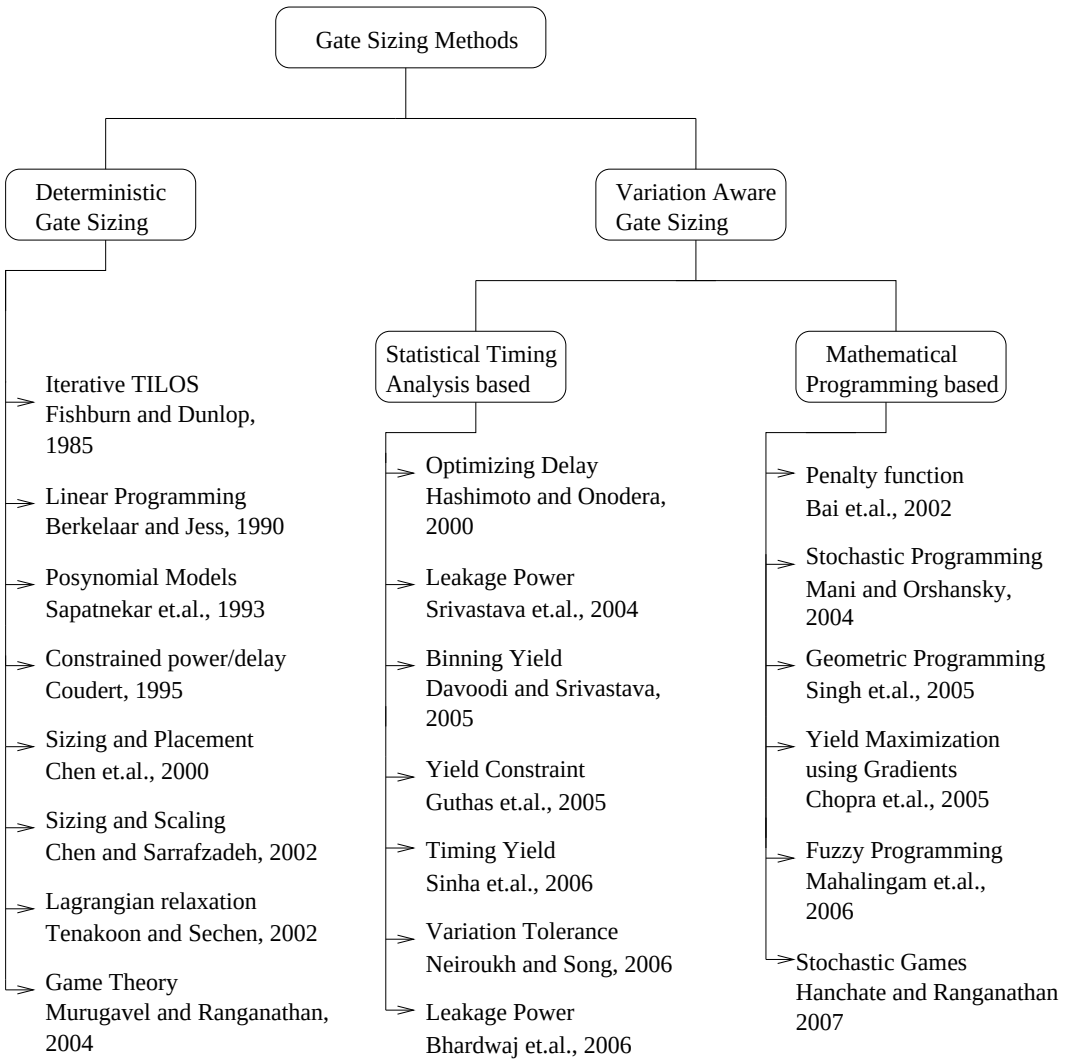


Figure 3.1 Taxonomy Diagram of Optimization Methods for Gate Sizing

An SSTA engine is used in the iterative optimization framework [49] to find the most critical gates to size in terms of power/delay sensitivity. A stochastic programming approach with chance (probabilistic) constraints is used in [53] and [54] to incorporate yield in the gate sizing problem formulation. However, the SSTA based approaches [42,49,95] use continuous distributions, which require a number of operations to be performed iteratively at each node and hence, involve higher runtimes [20,30]. The stochastic programming based statistical optimization technique, on the other hand, is reasonably fast, but is claimed in [40] that it can produce less optimized solutions compared to fuzzy programming, when tested with Monte-Carlo simulations.

In this chapter, we propose a new variation aware gate sizing algorithm considering the uncertainty due to process variations using the concept of fuzzy linear programming. In the context of fuzzy set theory, imprecision is defined as an uncertainty where it is difficult to even predict the average behavior of the outcome. Probability theory can be used to model situations in which the average behavior is predictable (situations that obey the law of large numbers) and enough information is available to model the probability distribution functions. The theory of fuzzy sets and systems on the other hand, has been used to model imprecision in different applications such as vision and robotics [64]. In VLSI design automation, fuzzy logic has been applied to model imprecise coefficients in VLSI testing and scheduling in high level synthesis [30]. To the best of our knowledge, this is the first time the concepts of fuzzy sets and systems and fuzzy mathematical programming is being used to model the uncertainty due to process variations in nanometer VLSI circuits. For simplicity, we use linear delay models [17] and linear membership functions [28]. However, more complex models including nonlinear or other posynomial models [56] can be easily incorporated into the fuzzy optimization flow.

The fuzzy optimization is a two step process. Initially, a deterministic optimization is performed assuming the worst and the average case values for the variation parameters to identify the bounds of the uncertain problem. The solution bounds and a variation parameter λ is used to transform the uncertain fuzzy problem into a crisp nonlinear programming problem [28, 64]. The term crisp problem in fuzzy mathematical programming context refers to a non-fuzzy or non-interval based real value. The variation parameter λ in this crisp problem ranges from (0,1) and implicitly models the interval values (fuzzy variables) of the original problem. This transformation is referred to as symmetric relaxation [28]. The additional parameter λ introduced during the transformation implicitly captures the uncertainty due to process variations and a maximization of this variable leads to high process variation resistance. Fuzzy numbers with nonlinear membership functions can be modeled by replacing this linear parameter λ with a nonlinear function in terms of the variation parameter. The solution of this crisp nonlinear problem represents the optimal value in the presence of variations.

The crisp problem, in general, has been proven to provide the most satisfying solution in the presence of variations in the coefficients of the constraints or objective function in the optimization [43, 56]. In the context of variation aware circuit optimization, the above crisp model with delay and power as constraints, can be used to maximize the robustness, i.e., the variation resistance of the circuit and thus the yield. The proposed approach has been tested on ITC'99 benchmark circuits

and the results indicate sizable savings in power compared to the worst case deterministic gate sizing approach. The proposed fuzzy programming based gate sizing also provides better results than the stochastic programming based gate sizing approach, [54], in terms of power savings with a comparable runtime. The results are validated using Monte-Carlo simulations, which indicate a high timing yield for the circuits designed with fuzzy gate sizing methodology. The rest of the chapter is organized as follows. In Section 3.2, we discuss the motivation as to why fuzzy programming is suitable for variation aware gate sizing. The details of the proposed modeling and the methodology for fuzzy gate sizing approach is given in Section 3.3, followed by experimental results and conclusions in Sections 3.4 and 3.5 respectively.

3.2 Why Fuzzy Programming for Variation Aware Gate Sizing ?

In this section, we discuss why fuzzy mathematical programming is well suited for modeling the uncertainty due to process variations in VLSI circuits. The impact of process variations in the nanometer-era are completely non-deterministic and the degree of uncertainty is expected to be worse in future generations [95]. A common approach to handling the uncertainties due to process variations has been to use probabilistic models, in which the uncertain parameters are represented in terms of probability distributions. However, the probabilistic way of evaluating and optimizing the uncertainties is computationally expensive due to the need for complicated multiple integration techniques needed for continuous distributions [20, 30] or due to the large number of scenarios for the corresponding discrete representation. Variation aware gate sizing, proposed in [49], using statistical static timing analysis requires very high execution times.

Furthermore, certain probabilistic modeling requires exhaustive description of uncertain parameters to build probabilistic distributions from historic (empirical) data. When such a description is not available (for example, in a new technology or for a new variation parameter, where extensive details of uncertainty is not known), we do not have enough information for deriving or obtaining the probabilistic distributions. Also, exhaustive Monte-Carlo simulations are needed to generate probability distributions for all the varying parameters. An alternative treatment of uncertainty is needed in the situations, where in an expert can predict or obtain only the mean and worst case values of an uncertain

parameter. Fuzzy mathematical programming and interval arithmetic can be used to make decisions in the above conditions.

In addition to the above arguments, Buckley has also shown in [40] that fuzzy programming based optimization guarantees solutions that are better or at least as good as their stochastic counterparts. The author provides a comparison of the stochastic and fuzzy programming methodologies using Monte-Carlo simulations. The fuzzy optimization, in uncertain environments, finds the best solution (supremum operation over all feasible solutions) as opposed to averaging (integrals over all feasible solutions) in stochastic programming based optimization. Hence, fuzzy programming selects a solution which is better than or at least as good as the stochastic solution. The above arguments provided us the motivation to investigate fuzzy mathematical programming approach to model uncertainty due to process variations in VLSI design automation. The performance of the proposed algorithm is compared with that of the stochastic programming based gate sizing in order to illustrate the efficiency of fuzzy programming for optimizing in presence of variations. It is shown in Section 3.4, that the proposed approach yields better power savings than stochastic gate sizing with comparable execution times under the assumptions of same models, setup, parameters and objective function for their implementations.

In the context of fuzzy gate sizing, the uncertainty due to process variations can be modeled as normally distributed random variables with mean E and standard deviation σ. We model these variations as interval valued fuzzy numbers in the range E-3σ and E+3σ, instead of using normal distribution,. The 3σ, value is assumed to be the deterministic worst case variation value, meaning all uncertain process parameters are set to 3σ, for maximum timing yield in worst case deterministic optimization. In the context of fuzzy mathematical programming, possibilistic distributions $\pi(.)$ are analogous to linear membership functions [30]. The triangular fuzzy number in Figure 2.1 is usually denoted by a triple X = (x^m, x^l, x^u), where x^m is the most possible value or the mean value and x^l, x^u are the lower and upper bounds, denoting the pessimistic and optimistic value of the number. Depending on the context, the value x^l can be pessimistic or optimistic variation from the mean value x^m and the same holds for the value x^u. In the context of VLSI circuit optimization, the triple $L_{eff} = (L^m_{eff}, L^l_{eff}, L^u_{eff})$ can be used to model variations in channel length. Since the general objective in circuit optimization is to minimize delay or power, the pessimistic value in this context for effective channel length is L^u_{eff}, which is the sum of $L^m_{eff} + 3\sigma$. Similarly if we model the gate oxide thickness as a fuzzy triple, the

pessimistic value is once again the upper bound value. In the next section, we explain the modeling and formulation of the variation aware gate sizing problem.

3.3 Proposed Variation Aware Fuzzy Gate Sizing

In this section, we describe our formulation of the gate sizing problem in the presence of uncertainty due to process variations. The problem of gate sizing can be defined as finding the optimal drive strengths such that the specified critical path timing is met and the overhead (power in this work) is minimized. We determine the size of the gates with the goal of minimizing dynamic power with delay as constraints. We use linear programming due to simplicity of modeling, faster execution time and availability of well developed fuzzy linear programming techniques for variation aware optimization [43]. However it should be noted that, fuzzy programming based optimization can also solve optimization problems in a nonlinear programming setup [56]. Next, we present the power and delay models used in this work.

3.3.1 Power and Timing Models

The dynamic power consumption of a gate (i) is given as,

$$P_i = \frac{1}{2} f V_{dd}^2 E_i (C_i + C_{wire}) + P_{sc} \tag{3.1}$$

where, P_i is the total dynamic power consumed by gate i, f is the clock frequency, V_{dd} is the supply voltage for the gate, E_i is the average switching activity of the gate, C_i is the intrinsic gate capacitance internal to the gate and C_{wire} is the sum of all the interconnects that fan-out from gate i. Thus, reducing the size (s_i) of the gate reduces the intrinsic gate capacitance of gate i, power consumption and fan-in load capacitance of the gate. Secondly, in this work we model gate delay as a linear function of gate size. The linear delay model proposed in [17], is given by

$$d_i = a_i - b_i s_i + c_i \sum_{j \varepsilon fo(i)} s_j \tag{3.2}$$

where, s_i refers to the size of gate i, $fo(i)$ is the set of gates that fan-out from gate i, constant coefficients a_i, b_i, c_i are empirically determined by extensive SPICE simulations for each gate in the library

for various sizes and fan-out counts. A similar nominal delay model has also been used in a recent stochastic programming based statistical gate sizing approach [54].

3.3.2 Variation Modeling with Spatial Correlation

The increasing influence of process variations on circuit yield is making variation aware optimization a requirement early in the design flow. The uncertainty due to process variations has been modeled in most works using the following equation,

$$D = d_i + \sum_{j=1}^{n} d_j X_j + d_r X_r \tag{3.3}$$

where, d_i is the nominal delay and X_j and X_r are the random parameters representing correlated and independent variations respectively. The magnitude of these variations is given by the variables d_j and d_r, which is determined from extensive simulations. The correlations of different areas of transistors are different and are usually high for gates close to each other in the die. The correlations in this work were included in the delay model as proposed by the authors in [4]. The modeling of spatial correlations by dividing the chip into regions and levels is shown in Figure 3.2. The lowest level (Level 0) of the die is divided into sixteen regions and they are grouped into sub-blocks upper levels. The correlation of connected gates in the die is directly proportional to the number of common regions the gates share in levels 0 and 1. The gates placed closer to each other have a similar variation characteristic and hence high correlation. The magnitude of variation in a gate is determined by the count of its fan-out gates, placed in the same Level 0 region. A gate with all its fan-out gates in the same region is modeled to have a smaller variation value.

We capture these variations using the concept of fuzzy numbers. We assume the delay of the gate as an interval with lower and upper bound values. In other words, each gate's delay is now a triangular value (average, low, high), instead of a single discrete value. We focus on the intra-die variations, meaning, each transistor can have a different amount of variation. In the pioneering work on process variations by Sani Nassif [79], its been pointed out that in the absence of real statistical data on a process run it's reasonable to assume a variation parameter value of 25% on the delay due to process variations. The uncertainty in the delay values are transferred to the coefficients b_i and c_i of the linear delay model shown in Equation 3.2. Following the above assumption, in the works, reported

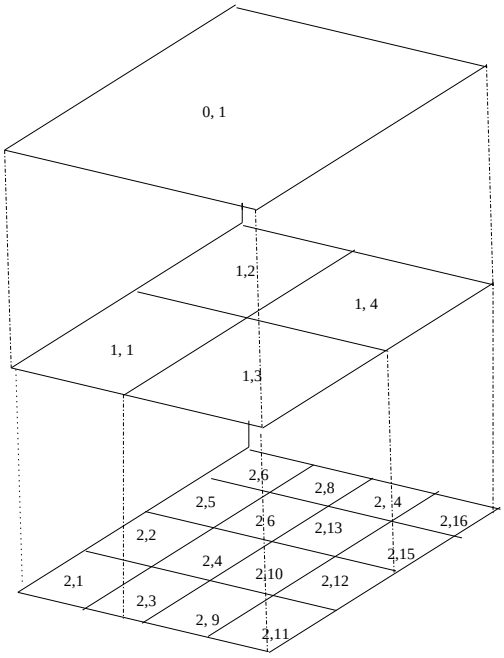


Figure 3.2 Spatial Correlation of Process Variations [Agarwal, 2003]

in [53, 54], it has been pointed out that the regression coefficients b_i and c_i closely approximate the variation effects of l_{eff} and t_{ox} based on simulation experiments with statistical data. They observed that the regression coefficients b_i and c_i to have a variation values of 8% and 10%, which in turn corresponded to the 25% variation effect in circuit delay. These coefficients are the fuzzy numbers of the triangular form with a linearly varying membership function. Since these observations were an outcome of significant experiments based on statistical data from real process runs, we followed the same assumptions in our work. This allows us to make a fair comparison with the work reported in [54]. Next, we explain the proposed fuzzy gate sizing approach for optimization in presence of process variations.

3.3.3 Variation Aware Fuzzy Gate Sizing

In this section, we use delay constrained dynamic power minimization for the gate sizing problem. If minimizing power is our sole interest, then all the gates can be set to minimum size. However, the problem objective is to achieve minimum power for a specified timing target. Hence, the cost function of the deterministic optimization formulation must include both delay and power. The deterministic formulation of the sizing problem is given by,

$$min \sum_i P_i \tag{3.4}$$
$$s.t. \quad D_p \leq T_{spec} \qquad \forall p \epsilon P$$
$$and\ D_p = \sum_{i \epsilon p} (a_i - b_i s_i + c_i \sum_{j \epsilon fo(i)} s_j)$$

where, T_{spec} is the specified timing target of the circuit, p denotes a particular path number in a circuit which belongs to the set of all paths P and D_p is the sum of the delays of all gates in path p. The summation of the dynamic power of all the gates is used as as the objective function. The dynamic power model in Equation 3.1 is substituted here. The fuzzy version of the above deterministic optimization problem with uncertain parameters is given by,

$$min \sum_i P_i \tag{3.5}$$
$$s.t. \quad D_p \leq T_{spec} \qquad \forall p \epsilon P$$
$$and\ D_p = \sum_{i \epsilon p} (a_i - \tilde{b}_i s_i + \tilde{c}_i \sum_{j \epsilon fo(i)} s_j)$$

where, s_i is bounded by minimum and maximum gate size, the coefficients $\tilde{b}_i$ and $\tilde{c}_i$ are the uncertain parameters. The uncertain parameters are modeled as fuzzy number triples of the form $(b_i, b_i - g_i, b_i + g_i)$ and $(c_i, c_i - h_i, c_i + h_i)$, where g_i and h_i are the maximum variations for the coefficients b_i and c_i respectively. The coefficient b_i and c_i closely approximate the variation in effective channel length (L_{eff}) and oxide thickness (t_{ox}). The authors in, [54], also follow a similar modeling for gate sizing in the presence of uncertainty using chance constrained programming. The fuzzy gate sizing problem is then transformed into a crisp nonlinear problem using the following steps. A deterministic optimization is performed initially with the varying coefficients set to worst and average case values of the fuzzy number. In the worst case optimization, the fuzzy gate delay equation in the fuzzy problem is replaced with the following equation.

$$d_i = (a_i - (b_i - g_i) s_i + (c_i + h_i) \sum_{j \epsilon fo(i)} s_j) \tag{3.6}$$

The gate delay in the above equation is the most pessimistic estimate, resulting in the worst possible delay for the gate. It can also be seen that the worst case estimate corresponds to the lower bound in coefficient b_i, since b_i is inversely proportional to the effective channel length and upper bound in coefficient c_i as it is directly proportional to gate oxide thickness. Similarly, the typical or nominal case of the gate delay is the case where the fuzzy numbers are fixed to their average case values. In the nominal case optimization, the fuzzy delay equations in the fuzzy problem is replaced with the following equation.

$$d_i = (a_i - (b_i)s_i + (c_i) \sum_{j \varepsilon fo(i)} s_j) \tag{3.7}$$

The deterministic optimization problem (Equation 3.4) is solved with the delay equations set to the worst case and nominal case equations. The KNITRO optimization solver available through the NEOS optimization server is used to solve the linear programming problems. The results of these optimization correspond to worst case gate sizing (wc_{sizing}) and nominal case gate sizing (nc_{sizing}) values. The above values and a new variation parameter λ are used to transform the fuzzy optimization problem into a crisp nonlinear programming problem using the symmetric relaxation method [64]. The crisp nonlinear problem for gate sizing in the presence of process variations is given by,

$$\begin{gathered} maximize \ \ \lambda \\ \lambda(nc_{sizing} - wc_{sizing}) - \sum_i P_i + wc_{sizing} \leq 0, \\ s.t. \quad D_p \leq T_{spec} \qquad \forall p \varepsilon P \\ and \ D_p = \sum_{i \varepsilon p} (a_i - (b_i - g_i * \lambda) s_i \\ + (c_i + h_i * \lambda) \sum_{j \varepsilon fo(i)} s_j) \end{gathered} \tag{3.8}$$

where, the parameter λ is bounded by 0 and 1. Even though, the parameter λ can take any values between 0 and 1, for the gate sizing problem, it can be easily bounded to a smaller value. In this work, we bound the λ value to be between 0.5 and 0.75. We estimated that such a smaller bound is sufficient due to the dual requirement of high yield and low overhead for the gate sizing optimization in presence of variations. The smaller bound speeds up the fuzzy gate sizing procedure by 2-3 times, without affecting the final solution. The crisp optimization problem has three variables, power (P_i), delay (D_p) and process variations (λ) in the above formulation. The parameter λ is the variation resistance (ro-

bustness) property of the circuit, meaning the ability to meet the timing constraint even in the presence of variations. The problem tries to maximize variation resistance, constraints delay value even with variations to be less than specified timing, and bounds the power value to be in between wc_{sizing} and nc_{sizing} values. One can favor the power value to be close to the nc_{sizing} value by maximizing the variation resistance value. Hence, the crisp optimization problem tries to satisfy all the three requirements to the maximum degree. It has been shown for problems in other application domains that the above formulation provides the most satisfying optimization solution in the presence of uncertainty [43, 56].

Another issue in the above optimization formulation is that the the number of paths in the circuit grows exponentially in the number of gates. Hence, the path based formulation is converted to a node based optimization problem [53, 54]. The node based formulation is a widely used technique [1, 13, 42, 53, 54, 95] to improve the computational efficiency of optimizing large circuits. The gate sizing problem with the node based formulation can be shown as,

$$
\begin{aligned}
& min \sum_i P_i \qquad (3.9)\\
s.t. \quad & a_j \leq T_{spec} \qquad \forall j \ \varepsilon \ input(PO)\\
s.t. \quad & a_j + D_i \leq a_i \qquad \forall i \ and \ \ j \ \varepsilon \ input(i)\\
& and \ D_i = (a_i - b_i s_i + c_i \sum_{j \varepsilon fo(i)} s_j)
\end{aligned}
$$

Here a_j is the arrival time at node j and T_{spec} is the timing specification. In the node based approach, the path based constraints are broken by using arrival time variables at each node. The number of such constraints is linear and is proportional to the number of interconnects in the circuit. The node based formulation introduces some sub-optimality (decrease in power optimization in this work). However, the decrease is negligible for circuits with less than 20 levels of logic [54]. Since the trend in the nanometer era, is towards higher clock speeds and lesser levels of logic, we believe that the overall impact is very less and the computational benefit in terms of running time of the optimization justifies such a modification. In the next section, we present the simulation steps and the experimental results of the fuzzy gate sizing approach tested on ITC'99 benchmark circuits.

3.4 Experimental Results

The proposed fuzzy linear programming optimization for gate sizing was tested on ITC'99 benchmark circuits. The complete simulation flow is shown in Figure 3.3. First, the RTL level VHDL netlists are converted to structural level Verilog netlist using the synopsys design compiler tool. The gate level netlist is completely flattened to the basic gates in the standard cell library. The output Verilog file from the design compiler is then placed and routed using the cadence design encounter tool. The benchmark circuits are synthesized using the TSMC 90nm db, lef and tlf libraries. The placed and routed netlist (DEF file), library of cell delay information and the Verilog file are given as an input to a C script (*DEF2AMPL*), which converts the netlist into a AMPL based mathematical program format for power minimization using fuzzy gate sizing. AMPL is a widely used modeling language for large scale mathematical programming problems. The average switching activity in each line was calculated by simulating each of the benchmark circuits with 100,000 random vectors. The equation coefficients for power and delay models for the standard cell library cells in the TSMC 90nm libraries are characterized for various gate sizes and fan-outs using hspice simulations. The fit is justified as the rms error is less than 7% for a restricted range of gate sizes (1x - 4x). The DEF2AMPL script uses these delay equations to generate the linear programming models for the benchmarks with delay coefficients set to mean and the maximum possible variation (worst case). The maximum variation in gate delay is assumed to be 25% from its mean value [79]. This is translated into appropriate values for the coefficients b_i and c_i in the delay model. The linear optimization problems are solved using the KNITRO Solver available through the NEOS server for optimization. The *DEF2AMPL* script uses the results of these optimizations and generates a fuzzy nonlinear AMPL model. The fuzzy nonlinear optimization problem is also solved using the KNITRO solver to find the optimal gate sizes in presence of variations in gate delay. The proposed fuzzy sizing approach is compared with the stochastic programming based gate sizing under uncertainty [54]. The latter method was also implemented with the same setup, parameters and objective functions for fairness in comparison.

The power reduction achieved by the fuzzy sizing approach compared to worst case deterministic sizing and the stochastic programming approach is documented in Table 5.2. The worst case sizing results correspond to the delay coefficients set to their maximum variation case. T_{spec} corresponds to the minimum required delay for the circuit, obtained by unconstrained delay optimization. Optimized

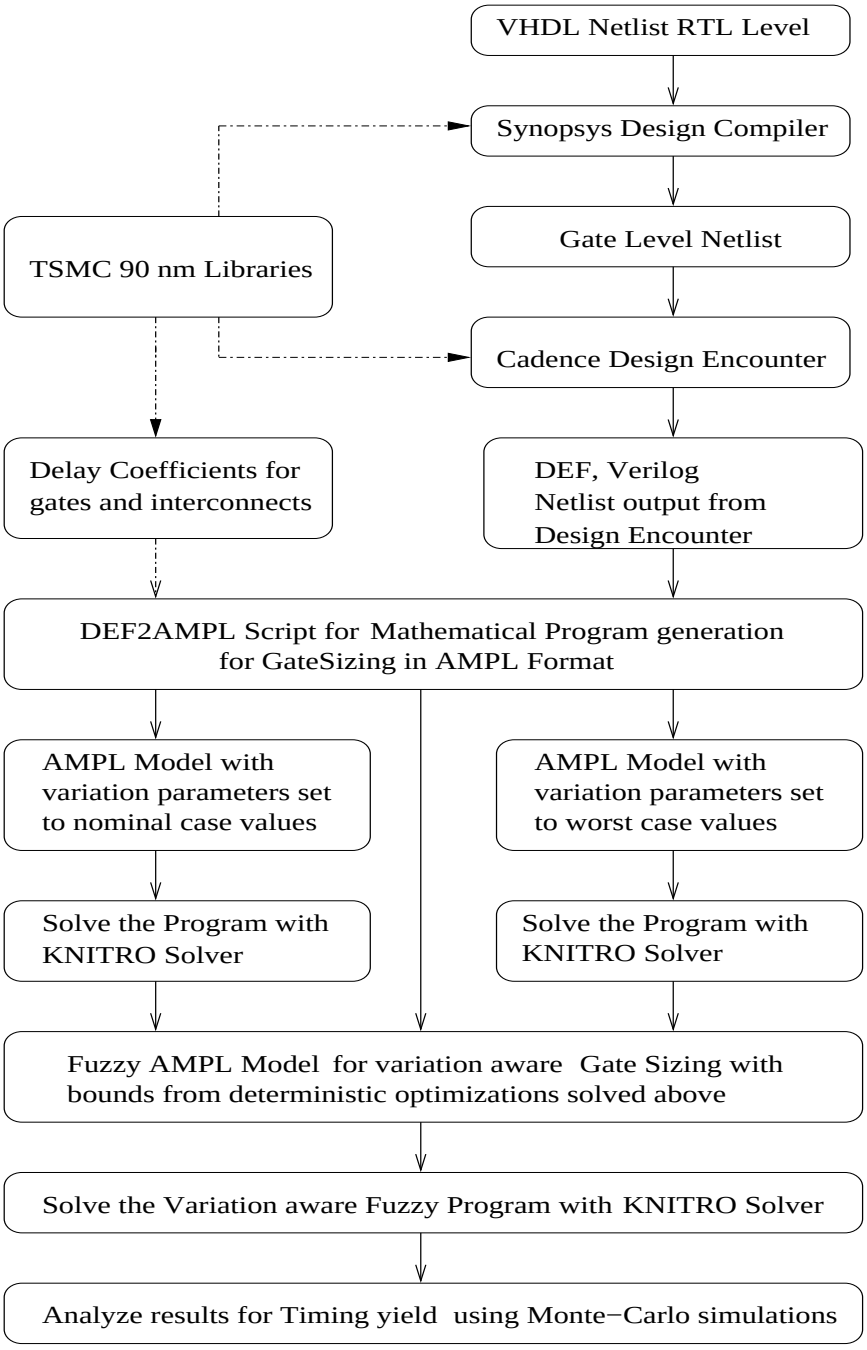


Figure 3.3 Fuzzy Gate Sizing: Simulation Flow

power values for worst case gate sizing, stochastic programming based gate sizing and fuzzy programming based gate sizing is shown in columns 4, 5 and 6 and the percentage reduction in power of fuzzy approach compared to worst case and stochastic approach is given in columns 7 and 8 respectively. The percentage savings compared to deterministic worst case sizing in power is calculated using the

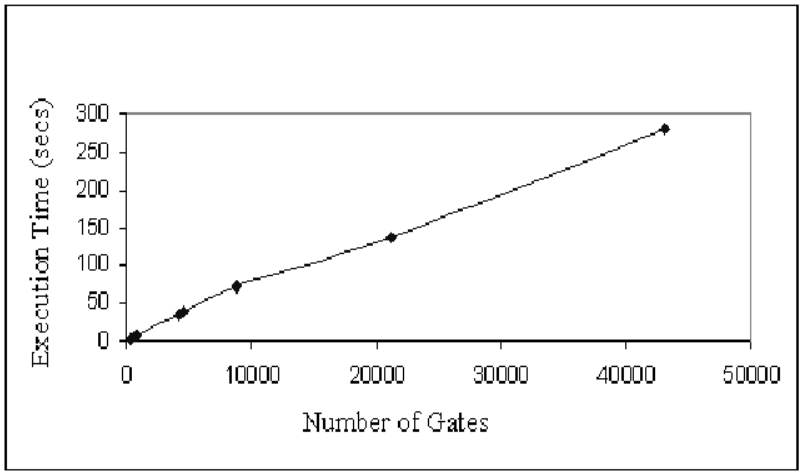


Figure 3.4 Fuzzy Gate Sizing: Execution Time

Table 3.1 Variation Aware Gate Sizing Results on Benchmark Circuits

ITC '99 Circuit	Number of gates	T_{Spec} (ns)	Gate Sizing Power(μw)			CPU time (sec)		% Reduction of F-GS over	
			DWC-GS	S-GS	F-GS	S-GS	F-GS	DWC-GS	S-GS
b11	385	0.25	288	254	232	1.54	1.62	19%	9%
b12	834	0.39	465	397	357	7.12	7.84	23%	10%
b14	4232	2.62	1826	1695	1524	29.18	34.45	16%	9%
b15	4585	2.98	1774	1521	1397	31.2	38.4	21%	8%
b20	8900	2.68	3797	3423	3120	68.14	72.13	18%	9%
b17	21191	3.48	8423	7812	7044	133.5	136.3	16%	10%
b18	43151	4.11	14176	13045	11753	269.55	282.36	17%	10%
Average Savings Percent								18.57%	9.2%
T_{spec} : Timing Specification; DWC-GS : Deterministic Worst Case Gate Sizing S-GS : Stochastic Gate Sizing [54] and F-GS : Fuzzy Gate Sizing [This work]									

following equation,

$$PR_1 = \frac{Power_{DWC-GS} - Power_{F-GS}}{Power_{DWC-GS}} \star 100 \tag{3.10}$$

Similarly, the percentage improvement of fuzzy sizing compared to stochastic sizing is calculated as,

$$PR_2 = \frac{Power_{S-GS} - Power_{F-GS}}{Power_{S-GS}} \star 100 \tag{3.11}$$

It can be seen that there is a sizable savings in power by using the fuzzy sizing approach as compared to deterministic worst case gate sizing and stochastic programming approach. The execution time of the fuzzy optimization approach is also shown in Table 5.2. We also studied the impact of our algorithm

on leakage power using the leakage models proposed in [15]. The fuzzy approach and the stochastic approach on the average improve leakage power by close to 17% and 8% respectively compared to the deterministic worst case (DWC) approach during variation aware gate sizing. It should be noted that the above result only indicates the impact of our proposed algorithm formulation on leakage power. However, a multiobjective optimization framework will need to be formulated to efficiently optimize both dynamic and leakage power.

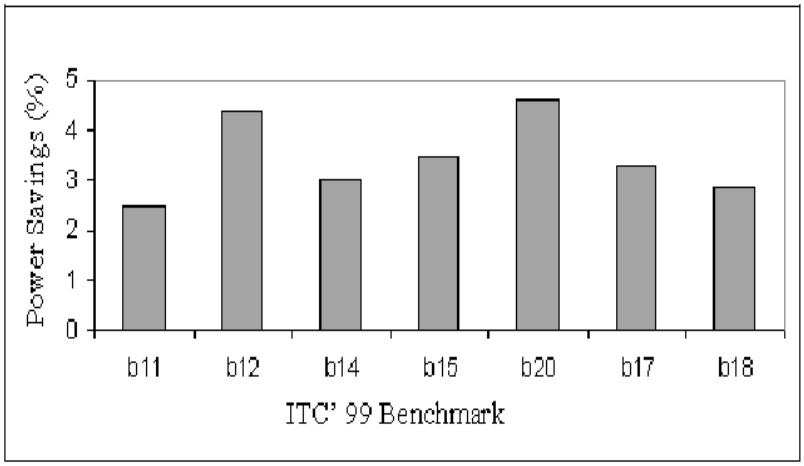


Figure 3.5 Improvement in Power Savings: F-GS With Correlations

The run-time of the fuzzy logic based optimization is comparable to the stochastic programming approach as seen from Table 5.2. Figure 3.4 also illustrates that the runtime complexity of fuzzy linear programming for gate sizing is close to linear in the number of gates in the circuit. Secondly, the effects of spatial correlation are considered as mentioned in Section 3.3. The variation magnitudes g_i and h_i of gate delay coefficients b_i and c_i are discretized and the contribution of each fan-out gate is weighed inversely with respect to the number of sharing regions between the gates. Figure 3.5 shows the percentage savings of the correlation aware variation modeling when compared to base fuzzy gate sizing approach. It can be clearly seen that the spatial correlation model eliminates further pessimism in the variation modeling and achieves a average savings of 3.4% power reduction compared to the base F-GS approach. Finally, to verify the result of the fuzzy sizing approach, we generated 10000 samples of the ITC'99 benchmarks. The circuits were fixed with gate size outputs from the fuzzy sizing method and the gate coefficient values b_i and c_i were assumed to have random variation value in the range b_i to $b_i - g_i$ and c_i to $c_i + h_i$ respectively. The variation value was generated from a uniform

distribution between these ranges. We then performed Monte-Carlo simulation with these random samples to determine the frequency of timing violations. The fuzzy logic approach had an timing yield of around 99-100% for all the benchmark circuits. This confirms the fact that the fuzzy gate sizing approach provides high resistance to process variations without compromising on the power overheads.

3.5 Conclusion

In this chapter, we proposed a new approach for gate sizing considering process variations using fuzzy linear programming. The variations in channel length and oxide thickness are modeled as fuzzy numbers with linear membership functions. The proposed fuzzy gate sizing approach maximizes variation resistance (robustness) of the circuit, with delay and power as constraints in the formulation. Experimental results on ITC'99 benchmark circuits indicate sizable savings in power and a runtime comparable with that of the stochastic sizing approach. The results validated using Monte-Carlo simulations, confirms the high variation resistance of the circuits sized using the fuzzy programming approach.

CHAPTER 4

VARIATION AWARE TIMING BASED PLACEMENT

4.1 Problem Definition

Circuit optimization techniques such as, gate sizing, incremental placement, buffer insertion, is commonly used to improve the performance of integrated circuits. Timing based incremental placement is crucial in nanometer circuits to meet the high performance requirement. The process finds the optimal locations of cells in a critical sub circuit such that the delay of the circuit is minimized. Circuit designers over the years, have used corner case models to optimize and analyze designs. The idea is to meet the timing specification at the best, worst and typical case model values. However, with process variations, the above test results can be far from the actual values. A guarded approach in terms of yield, to eliminate the effects of variability, is to perform deterministic optimization at the worst case values of the varying parameters. The worst case approach guarantees high timing yield, but leads to sub-optimal solutions in terms of performance. Timing yield in this context, is defined as the percentage of chips meeting the timing specification. Typical case value, on the other hand, guarantees optimal solutions but can result in unacceptable timing yield. It is clear that new methodologies are needed, which can guarantee a high timing yield and at the same time provide a solution with a high power/performance ratio.

Several researchers have investigated the effects of variations in timing analysis and statistical design optimization. Static timing analysis was replaced with statistical static timing analysis (SSTA) [11, 74], where continuous distributions are propagated instead of deterministic values to find closed form expressions for performance in the presence of variations. Recently, variation aware gate sizing for improving power and area for an acceptable yield has been investigated in [49, 54, 87]. Thus, the consideration of process variations is important in the design and the optimization of circuits. In this paper, we propose the use of fuzzy mathematical programming (FMP) and stochastic chance constrained programming for variation aware timing based incremental placement problem. The un-

certainty due to process variations are modeled using fuzzy numbers in the FMP case and using probabilistic constraints in the chance constrained programming formulation. Recently, the authors in [5, 81, 84] have considered process variations, while solving the placement problem. The authors in [81], have considered the effects of variations during placement in FPGAs. Variations due to lens aberrations have been considered in [5] and a fuzzy optimization flow for timing variations in [84, 86]. A taxonomy of related works on deterministic and variation aware timing based placement works are shown in Figure 4.1.

Taxonomy of Timing based Placement Works			
Type	**Methodology**	**Authors**	**Year**
Deterministic Timing Based Placement	Min−Max Timing Minimization	Kahng et.al.	2002
	Linear Programming based Timing Minimization	Chow and Bazargan	2003
	Net−weighted Interconnect length Minimization	Ren et.al.	2004
	Path based Accurate Timing Minimization	Chowdhary et.al.	2005
	High Performance LP based Timing Minimization	Luo et.al.	2006
	Monotone Ordering based Timing Placement	Hwang and Pedram	2006
Variation Aware Timing based Placement	Lens Abberation Aware Timing Variation Reduction	Kahng et.al.	2006
	FPGA Placement with Timing Variations	Srinivasan and Narayanan	2006
	Minimize Timing Variation with Fuzzy Programming	Mahalingam and Ranganathan	2007
	Minimize Timing Variation with Fuzzy and Stochastic Programming	Mahalingam and Ranganathan	2008

Figure 4.1 Taxonomy Diagram of Timing Based Placement

The problem of timing based incremental placement is an important part of the timing convergence flow. It can be formally defined as the process of finding the optimal locations of cells in a critical sub circuit such that the delay of the circuit is minimized. In timing based placement, the length of interconnects in the critical paths need to be minimized by changing the locations of certain cells [83, 91]. The timing of a circuit is usually measured in terms of the worst negative slack and the total negative slack. Slack in this context, is defined as the difference between the required time and actual arrival time of the signal. Timing driven placement approaches can be categorized into net-based [83, 97] and path based [6, 9, 19] approaches. The net-based approach translates the timing requirements into sensitivity coefficients of timing critical nets and performs a weighted wire length minimization. Hence, modeling the effects of process variations in these net-based approaches is not straightforward. On the other hand, the path based approaches hold an accurate timing view and minimize critical path delay more directly by involving path delay constraints in the optimization problem. A problem with the path based approach is their high computational complexity due to the exponential number of paths. But path based delay constraints can be transformed into node-based constraints [9, 87] to improve the feasibility of optimizing large circuits. The transformation only introduces a sub-optimality of 1-2% [54].

The theory of fuzzy sets and systems over the years, has been applied in VLSI design automation for high level synthesis [30] and for modeling variations in gate sizing [87]. The uncertainty due to variations can be modeled using fuzzy numbers with linear membership functions. The proposed timing based placement approach is formulated to minimize the worst negative slack of the circuit in the presence of process variations. The fuzzy optimization approach, starts with a deterministic optimization assuming the worst and the average case values for the variation parameters. The results of these deterministic optimizations are used to convert the fuzzy optimization problem into a crisp nonlinear problem using the symmetric relaxation method [64]. The crisp problem formulation, in general, has been shown to provide satisfactory solution in the presence of imprecision or variations in coefficients of the constraints or objective function in the optimization problem [56]. We show that the fuzzy optimization approach improves the variation resistance of the circuit without compromising on the achievable performance. The stochastic chance constrained programming (CCP) approach is again a well established technique for performing uncertainty aware optimization. It has previously been applied to model process variations during the gate sizing problem [53, 54]. Here, we also perform

variation aware nonlinear timing based placement using stochastic CCP. The stochastic CCP is cast as a robust mathematical program with varying parameters in the constraints of the formulation. The proposed approach uses probabilistic constraints to capture the uncertainty due to process variations. The optimization as a pre-processing step, converts these probabilistic constraints into an equivalent second-order conic program (SOCP) by explicitly using the mean, variance and the inverse-distribution of the varying parameters. Similar to the crisp-fuzzy problem, the translated stochastic-SOCP is solved using an interior point nonlinear optimization solver.

The rest of the paper is organized as follows. In Section 4.2, we motivate why FMP and SOCP are well suited for variation aware optimization. The proposed fuzzy timing based placement and Stochastic placement techniques are presented in Sections 4.4 and 4.5 respectively. The experimental results are presented in Section 4.6 and the conclusions in Sections 4.7.

4.2 Motivation

Timing based, incremental placement improves circuit delay by decreasing the length of the nets in the most critical paths, which are not identified in the global placement flow. Several researchers have investigated the timing based placement problem with deterministic models. However, with the increasing impact of variability in process parameters there is a strong need to develop optimization approaches with non-deterministic models. Probability density function and cumulative distribution function based techniques have been commonly used to perform uncertainty aware optimization. However, the probabilistic way of propagating and optimizing these uncertainties is computationally expensive due to the requirement of complicated multiple integration techniques needed for continuous distributions [20, 30]. The discrete probabilistic representation, on the other hand, can have huge execution time due to the large number of scenarios, which needs to be considered. Secondly, the problem of timing based placement is inherently suited to mathematical programming based optimization formulation. Hence, we investigate fuzzy mathematical programming and stochastic chance constrained programming based techniques for uncertainty aware optimization.

Further, both Stochastic and fuzzy programming based techniques, have been widely used to optimize uncertainty in several engineering areas. Modeling process variations using these techniques only requires the mean and variance values of the uncertain parameters. Further, Buckley [40] used

Monte-Carlo simulation, to show that fuzzy programming based optimization guarantees solutions that are better or at least as good as their stochastic counterparts. In [87], it is shown that fuzzy programming based optimization, can produce better solutions compared to their stochastic chance constrained programming technique. However, the above results were obtained using linear constraints and objective functions in the optimization formulation. In this work, we investigate a nonlinear formulation of both methods for the timing based placement problem.

4.3 Incremental Timing Based Placement

In this section, we initially explain the deterministic path based timing based placement formulation used as the basis in this work. Next, we discuss the fuzzy modeling of variations, the fuzzy mathematical programming formulation as well as the stochastic formulation for the timing based placement problem. The variation-aware timing based placement formulation is explained in detail in the context of fuzzy programming solution and only the necessary differences are highlighted in the stochastic placement subsection.

In timing based placement (TBP), the objective is to improve the performance by changing the locations of the critical cells. We use a critical cell selection algorithm, similar to the one proposed in [83] to identify the set of movable cells. The algorithm for the critical cell selection used in this work is shown in Figure 4.2. The algorithm marks cells with different critical id's (crit_id) based on their adjacency to the most critical path. Each critical cell is also marked with a movable distance, which is proportional to its crit_id and the move length. The move_length is estimated as a function of the number of gates placed in the neighborhood of the current cell. The problem of incremental TBP can be naturally modeled as a mathematical programming problem. The crit_id and move distance values set during the pre-processing stages are used as constraints and constants in the timing critical programming formulation. The variables in the timing based placement are the movable cell locations x_i and y_i and the interconnect boundary variables $left_j, right_j, top_j$ and bot_j. Here, we start with describing the changes in interconnect length and load capacitance and then show how these changes affect the delay of the circuit. Assume x_I and y_I to be the new locations of the cell I, then the half perimeter bounding box model for the net j introduces the following constraints.

Input : Placed and Routed Design: DEF, Verilog, TARPT and SPEF Files
Output: Critical Id and movable distance for each cell
for all gate *I* **do**
  Set crit_id[i] to 4;
end for

Perform timing analysis and identify gates and nets in the most critical path
for all gate *I* ε Critical Path **do**
  Set crit_id[i] to 1;
  for all ((gate *j* ε fanin(gate I)) or (gate *j* ε fan-out(gate I))) **do**
    Set crit_id[j] to 2;
  end for
  for all ((gate *k* ε fanin(gate j)) or (gate *k* ε fan-out(gate j))) **do**
    Set crit_id[k] to 3;
  end for
end for

Partition chip area into n squares ;
for all gate *I* **do**
  if I ε square j **then**
    Set move_length[i] to free_space[j] ;
  end if
end for

for all gate with crit_id[i] ≤ 3 **do**
  Mark them as movable in the optimization formulation;
  Set move_distance[i] to $\frac{move\_length[i]}{crit\_id[i]}$;
end for

Figure 4.2 Pre-Processing for Incremental Timing Based Placement

$$left_j \le x_i; right_j \ge x_i; \quad \forall \ cell \ \ I \ \ connected \qquad (4.1)$$
$$bot_j \le y_i; top_j \ge y_i \quad to \ \ interconnect \ \ j$$

where, I include all the cells connected to net j. The wire length of net j is then calculated as,

$$L_j = (right_j - left_j) + (top_j - bot_j); \qquad (4.2)$$

Table 4.1 Notations and Terminology

Symbol	Meaning	Symbol	Meaning
L_j	Half perimeter wire length of line j	Dg_i	Delay of gate I
x_i, y_i	x and y co-ordinates of cell I	Sg_i	Transition delay of gate I
Cp_j	Output capacitance of line j	S_i	Slew of net I
c	Unit capacitance value	$left_j, right_j$	Left and right co-ordinates of net j
r	Unit resistance value	top_j, bot_j	Top and bottom co-ordinates of net j
$cpin_j$	Pin capacitance of line j	$Dnet_j$	Delay of interconnect j
$Snet_j$	Transition delay on net j	λ	Variation resistance value
arr_i, arr_{gtj}	Arrival time on net I and gate j	Tspec	Timing Specification
η	Timing Yield of circuit	σ	Standard deviation
A_0, A_1, A_2	Fitting coefficients gate delay	B_0, B_1, B_2	Fitting coefficients transition delay
K_D, K_S	Elmore delay constants	V_{A1}, V_{A2}, V_{KD}	Variation Coeffients for A_1, A_2, K_D

The output capacitance Cp_j on net j is given by the sum of the wire capacitance on j and the pin capacitance $Cpin_j$.

$$Cp_j = c.L_j + Cpin_j; \tag{4.3}$$

where, c is the constant denoting the unit capacitance value. The gate delay and transition function are linear functions of the output load capacitance Cp_i and the input slew S_i.

$$Dg_i = A_0 + A_1.S_i + A_2.Cp_i; \tag{4.4}$$

$$Sg_i = B_0 + B_1.S_i + B_2.Cp_i; \tag{4.5}$$

where, the constants A_0, A_1, A_2, B_0, B_1 and B_2 are the fitting coefficients for different gates in the library. Also, these coefficients differ for different inputs of a gate and depending on the falling or a rising transition. Finally the interconnect delay and transition time on net j is given by,

$$Dnet_j = K_D.r.L_j.(\frac{c.L_j}{2} + Cpin_j); \tag{4.6}$$

$$Snet_j = K_s.r.L_j.(\frac{c.L_j}{2} + Cpin_j); \tag{4.7}$$

where, r is the unit resistance and K_D, K_s are the Elmore delay constants 0.69 and 2.2 respectively. A complete list of the notations and terminology in the equations of timing based placement, fuzzy and stochastic programming context is given in Table 4.1. The above equations for interconnect and gate

delays denote the base (nominal) delay used in this work. The uncertain parameters in accordance with, recent statistical design optimization works, can be modeled as:

$$D = d_i + \sum_{j=1}^{n} d_j X_j + d_r X_r \tag{4.8}$$

where, d_i is the nominal delay and X_j and X_r are the random parameter determining correlated and independent variations respectively. The magnitude of these variations is given by the variables d_j and d_r, which is determined from extensive simulations. In this work, we capture these variations using the concept of fuzzy numbers and probabilistic constraints. We assume that the delay of a gate and interconnect as a value with upper and lower bounds. In other words, each gate's delay is now a triangular value of the form (average, low, high). The spatial correlations of different areas of transistors are different and captured by dividing the circuit area into n regions as approached in [35]. Further, we model the uncertainty in the gate delay equation due to the variations in the net length using the fitting coefficients A_1 and A_2. These coefficients are modeled as a triangular value with higher and lower bounds representing the deviation from the nominal value. Secondly, the uncertainty in the interconnect delay is modeled by abstracting the Elmore delay constant (K_D) to be an interval value. These coefficients are of triangular form with a linearly varying membership function. Next, we explain the proposed fuzzy timing based placement approach for optimization in presence of process variations.

4.4 Variation Aware Fuzzy Timing Based Placement

The concept of fuzzy sets was introduced by Zadeh in 1965, where an element's membership in a set could be any value within the range [0, 1]. Along with fuzzy sets, Bellman and Zadeh [64] also introduced fuzzy programming based optimization. Since then, several models and approaches have been proposed in the literature for uncertainty management which is based on fuzzy linear programming, fuzzy multi-objective programming, fuzzy dynamic programming, fuzzy integer programming, possibilistic programming and fuzzy nonlinear programming [56]. The reader is referred to Sakawa [56] and Zadeh [64] for detailed treatment of the basics on fuzzy mathematical programming (FMP). An important step in fuzzy programming is the modeling of uncertainty. In this work, we model the uncertainty due to process variations using fuzzy numbers with linear membership func-

tions. In recent statistical optimization works [49, 74], these variations are modeled as normally distributed random variables with zero mean and standard deviation σ. In this work, we extract the values of mean and 3σ to model these variations as interval valued fuzzy numbers. The 3σ value is assumed to be the deterministic worst case variation value, meaning all the random parameters are set to 3σ for maximum timing yield in the worst case bound deterministic optimization.

Fuzzy numbers are defined using possibilistic distributions. In the context of FMP, it is analogous to linear membership functions [30]. Triangular and trapezoidal and other nonlinear membership functions can be used for modeling uncertainty, while solving FMP problems. The nonlinear membership functions can be used for more accurate modeling [56]. For simplicity, we model the uncertainty due to process variations using triangular membership functions. The variations are represented using a triple X = (x^m, x^l, x^u), where x^m is the most possible value or the mean value and x^l, x^u are the lower and upper bounds. Depending on the context, the value x^l can be a pessimistic or optimistic variation from the mean value x^m and the same holds for x^u. In the context of VLSI circuit optimization, the triple $L_{eff} = (L^m_{eff}, L^l_{eff}, L^u_{eff})$ can be used to model the variations in channel length. The pessimistic value in this case is the upper bound L^u_{eff} which is the sum of $L^m_{eff} + 3\sigma$. Since the effective length is inversely proportional to our objective performance, the value $L^m_{eff} + 3\sigma$ is the upper bound or worst case variation value. The fuzzy programming solution methodology can be explained with a simple linear programming problem of the form,

$$\begin{aligned} &maximize \sum_{i=1}^{n} a_i x_i \\ &subject\ to \sum_{i=1}^{n} \tilde{b}_{ji} x_i \leq c_j, \quad 1 \leq j \leq m \end{aligned} \tag{4.9}$$

where, m is the number of constraints, n the number of variables, c_j is the limit of the constraint, b_{ji} and a_i are constant regression coefficients, x_i is the variable in the optimization formulation and at least one $x_i > 0$. In the above optimization problem, the coefficient $\tilde{b}_{ji}$ is the interval valued fuzzy number which has a mean value, b_{ji} and a maximum variation of d_{ji}. The upper bound is assumed to be the pessimistic variation for this fuzzy number. The fuzzy number $\tilde{b}_{ji}$ is also assumed to vary linearly with the value of the variable x_i. As a pre-processing step, the fuzzy programming problem is solved with

the variation parameters set to their worst and typical case values. The results of these deterministic optimizations and the symmetric relaxation theorem are used to convert the fuzzy program into a crisp program (Equation 4.10), which represents an optimal solution in the presence of variations. The conversion of the fuzzy program into a crisp program is also referred to as the de-fuzzification step.

$$
\begin{aligned}
&maximize \ \ \lambda \\
&\lambda(Obj_l - Obj_u) - \sum_{i=1}^{n} a_i x_i + Obj_u \leq 0, \\
&\sum_{i=1}^{n} (b_{ji} + \lambda d_{ji}) x_i - c_j \leq 0, \qquad 1 \leq j \leq m \\
&x_j \geq 0, \ \ 0 \leq \lambda \leq 1 \qquad 1 \leq i \leq n
\end{aligned}
\tag{4.10}
$$

Here, λ is referred to as the variation resistance parameter. The crisp problem is formulated as to maximize the variation resistance parameter and maintain the original optimization objective within the deterministic optimization bounds. The solution of the problem can be interpreted as representing an overall degree of satisfaction in the presence of varying parameters [56]. In the next section, we explain the fuzzy linear programming formulation for the timing based incremental placement problem in the presence of uncertainty due to process variations.

In this work, we perform timing minimization by changing the locations of the critical cells in the presence of variations. The problem minimizes the worst negative slack with the location constraints given in Equation 4.1 and the delay constraints given in Equations 4.4, 4.5, 4.6 and 4.7). The deterministic formulation of the incremental TBP problem can be shown as,

$$
\begin{aligned}
&minimize \ \ T_{spec} \\
&arr_{net} \leq T_{spec} \qquad \forall net \varepsilon EP \\
&arr_j = arr_{gti1} + Dg_{gti1} \ where \ j \varepsilon out(gti1) \\
&arr_{gti} \geq arr_{gti,k} + Dnet_{gti,k} \forall \ k \varepsilon ip(gti)
\end{aligned}
\tag{4.11}
$$

where, T_{spec} is the worst negative slack or the critical path delay of the circuit, EP is the set of end points of the circuit, which are the primary outputs in case of a combinational circuit and input to flip-flops in the case of a sequential circuit. The arr_netid and arr_gtid denote the arrival time variables

for each interconnect and gate, which are replicated for the whole circuit along with the gate and delay equations. The delay equation's Dg_i and $Dnet_j$ are expanded as shown in equations 4.4 and 4.6. In addition to the above constraints the location constraints (Equation 4.1) are also part of the optimization formulation, However not shown here, since it is not inspected by the variation aware formulation. The fuzzy version of the above deterministic optimization formulation with uncertain parameters is given as follows,

$$\begin{gathered} minimize \;\; T_{spec} \\ arr_{net} \leq T_{spec} \quad \forall net \varepsilon EP \\ arr_j = arr_{gti1} + \tilde{Dg}_{gti1} \; where \; j \varepsilon out(gt1) \\ arr_{gti} \geq arr_{gt,k} + \tilde{Dnet}_{gti,k} \; \forall \;\; k \varepsilon ip(gti) \end{gathered} \tag{4.12}$$

where, the coefficients $\tilde{Dg}_i$ and $\tilde{Dnet}_j$ are the uncertain parameters. The uncertain parameters are modeled as fuzzy number triples of the form $(Dg_i, Dg_i - m_i, Dg_i + m_i)$ and $(Dnet_i, Dnet_i - n_i, Dnet_i + n_i)$, where m_i and n_i are the maximum variations for the nominal gate delay Dg_i and interconnect delay $Dnet_i$ respectively. The fuzzy problem is then transformed into a crisp nonlinear problem using the following steps. The deterministic optimization is performed initially with the varying coefficients set to worst and average case values of the fuzzy number. In the worst case optimization, the gate delay equations in the fuzzy optimization problem are replaced with the following equation.

$$\begin{gathered} Dg_i = A_0 + (A_1 + V_{A1}).S_i + (A_2 + V_{A2}).Cp_i \\ Dnet_j = (K_D + V_{KD}).r.L_j.(\tfrac{c.L_j}{2} + Cpin_j) \end{gathered} \tag{4.13}$$

where, V_{A1}, V_{A2} and V_{KD} represents the variation values applied to these coefficients to represent the worst case variation in gate and interconnect delay. The gate and interconnect delay in the above equations are the pessimistic estimates, resulting in high delay value. Similarly, the typical or nominal value of the gate delay is the case where the fuzzy numbers are fixed to their average values. In the nominal case optimization, the fuzzy delay equations in the fuzzy problem are replaced with the

following equation.

$$Dg_i = A_0 + (A_1).S_i + (A_2).Cp_i; \tag{4.14}$$
$$Dnet_j = (K_D).r.L_j.(\tfrac{c.L_j}{2} + Cpin_j);$$

The deterministic optimization problem is solved with the delay equations set to these worst case and nominal case equations. KNITRO optimization solver available through the NEOS optimization server is used to solve these linear programming problems. The results of these deterministic optimizations correspond to the worst negative slack of the worst case timing setting (wc_{tbp}) and nominal case timing setting (nc_{tbp}). Using these bounds and a new variation parameter λ the fuzzy optimization problem is transformed into a crisp nonlinear programming problem using the symmetric relaxation method [64]. The incremental TBP problem in the presence of process variations is converted to its corresponding crisp formulation as shown in Equation 4.15.

$$maximize \ \ \lambda \tag{4.15}$$
$$\lambda(nc_{tbp} - wc_{tbp}) - T_{spec} + wc_{tbp} \leq 0$$
$$arr_{net} \leq T_{spec} \qquad \forall net \varepsilon EP$$
$$arr_j = arr_{gti1} + \tilde{Dg}_{gti1} \ where \ j \varepsilon out(gti1)$$
$$arr_{gti} \geq arr_{gti,k} + \tilde{Dnet}_{gti,k} \ \forall \ k \varepsilon ip(gti)$$
$$Dg_i = A_0 + (A_1 + V_{A1}.\lambda).S_i + (A_2 + V_{A2}.\lambda).Cp_i;$$
$$Dnet_j = (K_D + V_{KD}.\lambda).r.L_j.(\tfrac{c.L_j}{2} + Cpin_j);$$

where, the parameter λ is bounded by 0 and 1. The spatial correlations can be incorporated by making the coefficients V_{A1}, V_{A2} and V_{KD} in the delay equation as a function of the current and fan-out gate's location in the chip. The chip area can be partitioned into n areas as in [35] such that the gates within the same block will have high correlation. Even though, the parameter λ can take any value between 0 and 1, it can be easily bounded to a smaller value in the TBP. Here, we bound the λ value to be between 0.3 and 0.7. We estimated that such a smaller bound is sufficient due to the dual requirement of high

yield and high performance for the timing based placement optimization in presence of variations. The smaller bound improves the performance of the fuzzy optimization problem by a factor of 2-3x, without any effect on the solution optimality.

The crisp optimization problem has two variables in the cost function namely delay and variation parameter λ. The parameter λ is the variation resistance (robustness) property of the circuit, meaning the ability to meet timing constraint even in the presence of variations. The problem tries to maximize variation resistance and bounds the delay value to be in between wc_{tbp} and nc_{tbp} values. Favoring the delay value to be close to the nc_{tbp} value, as the objective is to maximize variation resistance of the circuit. It has been proved for problems in other domain that the above formulation provides the most satisfying solution for optimization in presence of variations [56]. Finally, a timing driven legalization is performed to remove cell overlaps in the circuit. In the next section, we explain the stochastic placement formulation, with only the necessary changes in the variation-aware modeling of the placement problem.

4.5 Stochastic Timing Based Placement

In this section, we describe our formulation of the stochastic timing based placement optimization technique. The formulation is cast as a robust mathematical program, which is then reformulated into an equivalent second order conic program (SOCP). The SOCP analytically captures the dependence of the constraints and objectives of the optimization using the mean and variance of the uncertain parameters. The advantage of stochastic formulation is the ability to consider the uncertainty of the constraints in an explicit fashion. The stochastic chance constrained programming technique models the uncertainty using probabilistic constraints. The probabilistic constraints are converted to an equivalent SOCP with mean $\bar{m}$ and standard deviation σ. The mean corresponds to nominal delay value as explained in the fuzzy modeling section and the standard deviation is chosen in accordance with the values in [79]. The SOCP formulation also has an inverse cumulative distribution function (cdf) function, which controls the yield of the optimization problem. The estimated timing yield of the optimized circuit is directly proportional to the value of the inverse cdf function. We use an inverse cdf value of 3 (corresponding to parameter 3σ in fuzzy formulation) to achieve a 99.7% timing yield. The main difference of the stochastic placement formulation compared to the fuzzy placement technique

described in the previous section is the arrival time constraints with gate delay (*Dg*) and interconnect delay (*Dnet*), which are assumed to vary due to process variations. The deterministic version of the timing based placement's arrival time constraints can be shown as,

$$arr_j \geq arr_{gti} + Dg_{gti} \;\; where \;\; j \varepsilon out(gti)$$
$$arr_{gti} \geq arr_{gti,k} + Dnet_{gti,k} \;\; \forall \;\; k \varepsilon ip(gti) \tag{4.16}$$

Here, the incremental interconnect arrival time constraint bounds the arrival time of an interconnect output (j) to be greater than or equal to the sum of arrival time of the gate (gti) and the gate delay (Dg_{gti}). The second constraint, bounds the arrival time of a gate (gti) to be greater than or equal to the sum of arrival time of its inputs $arr_{gti,k}$ and the interconnect delay $Dnet_{gti,k}$. A separate constraint is added to the formulation for each input k of gate (gti). The above two constraints are added for all the gate and interconnect of the circuit. The stochastic formulation for the above deterministic arrival time constraints can be shown as,

$$P(arr_j \geq arr_{gti} + Dg_{gti}) \geq \eta \;\; where \;\; j \varepsilon out(gti)$$
$$P(arr_{gti} \geq arr_{gti,k} + Dnet_{gti,k}) \geq \eta \;\; \forall \;\; k \varepsilon ip(gti) \tag{4.17}$$

The formulation is based on chance constrained programming, with roots in stochastic programming, in which the constraint has to be met with a probability of η. In the context of timing based placement, the parameter η corresponds to the timing yield of the circuit. An equivalent formulation for the above probabilistic constraint using mean or nominal value $\bar{D}$, cumulative distribution function (cdf(ϕ)) and standard deviation (σ)can be shown as,

$$arr_j \geq arr_{gti} + \bar{Dg}_{gti} + \phi^{(-1)}(\eta)\sigma_{Dg_{gti}} \;\; where \;\; j \varepsilon out(gti1)$$
$$arr_{gti} \geq arr_{gti,k} + \bar{Dnet}_{gti,k} + \phi^{(-1)}(\eta)\sigma_{Dnet_{gti,k}} \;\; \forall \;\; k \varepsilon ip(gti) \tag{4.18}$$

Here, the probabilistic constraint for gate delay is replaced with the sum of mean gate delay $\bar{Dg}_{gti1}$ and the product of its standard deviation $\sigma_{Dg_{gti}}$ and an inverse cdf value $\phi^{(-1)}(\eta)$ for high timing yield. Similarly, the constraint for interconnect delay is also replaced with its equivalent mean, cdf and standard deviation value. The above constraint is qualitatively different from the deterministic one, since it considers both the mean and variance of the delay values as the decision variables of the optimization problem. The complete stochastic CCP based timing based placement formulation can be shown as,

$$minimize \;\; T_{spec} \tag{4.19}$$

$$arr_{net} \leq T_{spec} \quad \forall net \varepsilon EP$$

$$arr_j \geq arr_{gti} + \bar{Dg}_{gti} + \phi^{(-1)}(\eta)\sigma_{Dg_{gti}} \;\; where \;\; j \varepsilon out(gti1)$$

$$arr_{gti} \geq arr_{gti,k} + \bar{Dnet}_{gti,k} + \phi^{(-1)}(\eta)\sigma_{Dnet_{gti,k}} \;\; \forall \;\; k \varepsilon ip(gti)$$

Input : Placed and Routed Design: DEF, delay coefficients, models
Output: Cell locations for Variation Aware Timing based Placement

1. Use input information to create worst case, nominal case, stochastic and fuzzy variation aware optimization problems ;
2. Optimize the worst and nominal case programming problems using KNITRO solvers and mark the objective values to wc_{tbp} and nc_{tbp} respectively ;
3. Using wc_{tbp} and nc_{tbp} values and a new variation parameter λ create a fuzzy program as shown in equation 4.15 ;
4. Use a variation value of 3σ as in worst case optimization by making timing yield constant set to a constant value 3 ;
5. Solve the fuzzy program and stochastic program (equation 4.19) using a KNITRO optimization solver ;

Figure 4.3 Process Variation Aware Incremental Placement

As mentioned in the previous section, In addition to the above shown timing constraints the location constraints (Equation 4.1) are also part of the stochastic optimization formulation. The SOCP formulation with mean and variance values can also be efficiently solved using an interior point optimization solver. It was estimated from simulations, that the inverse cdf value $\phi^{(-1)}(\eta)$ for high timing

yield can be substituted with a value of 3 for a predicted timing yield of 99.7%. A simple outline for the steps involved in the fuzzy and stochastic approaches is shown in Algorithm 4.3.

4.6 Experimental Results

In this section, we present the simulation flow and experimental results of the proposed fuzzy programming based timing placement and compare it with stochastic and worst case process variation approaches. The variation-aware placement approaches were tested on ITC'99 benchmark circuits. The complete simulation flow of the proposed fuzzy and stochastic approaches is shown in Figure 6.6. First, the RTL level VHDL netlists are converted to structural level Verilog netlist using the synopsys design compiler tool. The output Verilog file from the design compiler is then placed and routed using the cadence design encounter tool. The design encounter is also used to perform clock synthesis and timing analysis of the input netlist. The placed and routed netlist (DEF File), timing analysis report (TARPT) and the Verilog file is given as an input to a C script (*DEF2AMPL*), which converts the netlist into an AMPL based mathematical program format for timing based placement optimization. AMPL is a widely used modeling language for large scale mathematical programming problems. The *DEF2AMPL* script, on different options, generates the worst case deterministic, typical case deterministic, stochastic or the fuzzy version of the timing based placement problem. The *DEF2AMPL* script, as pre-processing step also generates the coefficients A_0, A_1, A_2, B_0, B_1 and B_2 using interpolation. The script selects the maximum allowable displacement for cells depending on the circuit area and the gate's criticality. The list of values for the interpolation is generated from the design encounter tool. The maximum variation in gate and interconnect delay is assumed to be 25% from the mean value due to the varying process parameters, which is in accordance with the results in, [12, 79]. This is translated to the appropriate values of the variation parameters $A_1, A_2 and K_D$, as mentioned in the previous section.

The mathematical programming problems are solved using the KNITRO nonlinear optimization solver available through the NEOS server for optimization. The results of the deterministic nominal and worst case optimizations are also fed to *DEF2AMPL* script for generating the bound's constraint in the fuzzy nonlinear AMPL model. The fuzzy and the stochastic optimization problem find the optimal

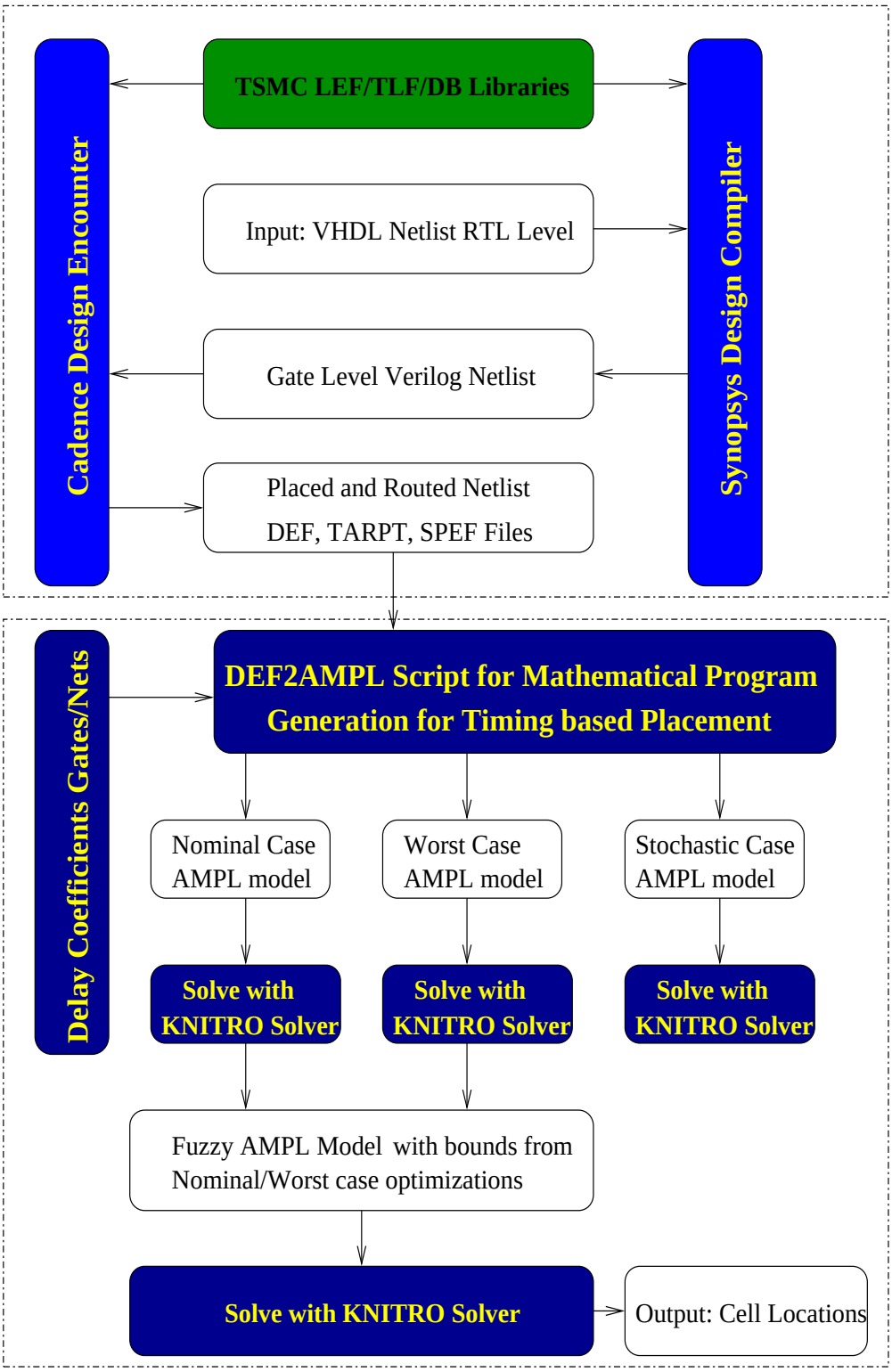


Figure 4.4 Variation Aware Timing Based Placement: Simulation Flow

Table 4.2 Variation Aware Placement Results on Benchmark Circuits

ITC' 99 Circuit	Total gates	Movable Cells	Worst Negative Slack (ns)			% Improvement	
			DWC-TBP	S-TBP	F-TBP	Fuzzy Vs WC	Stoc Vs WC
b13	309	42	0.121	0.106	0.104	14.1%	12.4%
b11	385	79	0.143	0.127	0.125	12.5%	11.1%
b12	834	98	0.210	0.189	0.185	11.9%	10.0%
b14	3651	1099	3.87	3.43	3.36	13.1%	11.3%
b15	6452	665	2.72	2.47	2.41	11.4%	9.19%
b20	8900	665	5.65	4.99	4.92	12.9%	11.6%
b22	12128	891	7.22	6.68	6.33	12.3%	7.47%
b17	21191	1280	8.61	7.64	7.52	12.6%	11.2%
Average Percent Timing Improvement						12.6%	10.53%
Legend - WNS: Worst Negative Slack; F-TBP: Fuzzy Timing Based Placement; Legend - DWC: Deterministic Worst Case; S-TBP Stochastic Timing Based Placement;							

cell locations in the presence of variations. Finally, timing aware cell legalization is performed to remove overlaps.

The timing improvement achieved by the fuzzy sizing approach compared to worst case deterministic sizing is documented in Table 5.2. The worst case placement results correspond to the delay coefficients set to their maximum variation case. The percentage improvement of fuzzy approach compared to deterministic worst case approach is calculated as,

$$PR_1 = \frac{WC - TBP - F - TBP}{WC - TBP} \star 100 \tag{4.20}$$

It can be seen that there is a savings of around 12% in worst negative slack by using the fuzzy placement approach as compared to deterministic worst case optimization. Secondly, we also present the results of the stochastic optimization framework in Table 5.2. The percentage improvement of stochastic placement approach compared to worst case setting is calculated as,

$$PR_1 = \frac{WC - TBP - S - TBP}{WC - TBP} \star 100 \tag{4.21}$$

It can be seen that the stochastic placement approach improves the timing by around 10% as compared to the deterministic worst case optimization at 99.7% timing yield level. As predicted by Buckley in [40], the fuzzy optimization approach outperforms stochastic programming techniques even with nonlinear constraints in the formulation. Finally, to verify the timing yield of the fuzzy based placement approach, we generate multiple samples of the ITC benchmark circuits. The sample

instances of the benchmark circuits are fixed with placement location outputs from the fuzzy method and the coefficients of delay are assumed to have random variation value. The variation value is generated from a uniform distribution between minimum and maximum variation values used in the optimization. We then performed Monte-Carlo simulation of these random instances to determine the frequency of timing violations, i.e., number of times delay of the random circuit is greater than specified timing (T_{spec}). The fuzzy logic approach had a timing yield of around 99-100% for all the benchmark circuits. This confirms the fact that the FMP is an efficient approach to design circuits with high yield without sacrificing much on performance.

4.7 Conclusion

In this chapter, we described a formulation for variation-aware timing based placement problem using fuzzy and stochastic approaches. The uncertainties due to process variations in these formulations are respectively modeled as fuzzy numbers and probabilistic constraints. The coefficients in the gate and interconnect delay arrival time constraints are assumed to vary in the optimization formulation. The proposed variation-aware timing based placement maximizes variation resistance (robustness) of the circuit, with the timing information represented as constraints. Experimental results on ITC' 99 benchmark circuits indicate a savings of around 12% for fuzzy programming and 10% for stochastic programming in average compared to the worst case deterministic approach. The proposed results validated using Monte-Carlo simulations also confirm high timing yield for circuits designed with the variation-aware techniques.

CHAPTER 5

VARIATION AWARE BUFFER INSERTION

5.1 Introduction

In nanometer era, it is crucial to consider area, power and process variation metrics in the optimization formulation. Further, interconnects have become longer and net delay has become more dominant than logic delay. The interconnect delay, to its first order, is proportional to the square of the length of the wire. This has increased the importance for interconnect driven performance optimization techniques such as, buffer insertion/sizing, wire sizing/spacing and driver sizing. Of these techniques, buffer insertion has effectively been able to divide the wires into smaller segments and bring the wire delay to almost linear in terms of its length. Further, it has also been estimated in [62], that 35% of the total standard logic cells in a circuit will be buffers at the 65nm technology level. Therefore, it is highly important to find optimal number of buffers for low overhead timing optimization.

Several researchers have proposed buffer insertion techniques and they can be mainly classified as net-based [16, 93], path based [23, 29] and network-based [31, 85, 96] techniques. In net-based approach, buffers are inserted in nets to create positive slack at the source. Even with criticality based net-ordering mechanisms, it may lead to sub-optimal over-buffering due to a lack of global view. The path based buffer insertion algorithms abstracted a path as a routing tree and inserted buffers on them to minimize the critical path delay [23]. The approach achieves more reduction in buffers costs compared to net-based approaches, but still suffers from lack of global view as it considers each path independently. Because of their greedy approach, earlier processed nets/paths can over-consume buffers resulting in a non-optimal solution [29]. Circuit wise buffer insertion techniques, on the other hand takes a whole circuit as an input instead of an individual net or path. The first such approach in, [31], uses largrangian relaxation techniques but suffers from unrealistic assumptions of at least one buffer in each interconnect. The network-based approach in [96] uses a piece-wise linear programming formulation to model the nonlinear delay improvements of the buffer insertion problem. However,

the above approaches do not consider variability in their formulation and hence are not suitable for optimizing designs in the nanometer regime.

Variation aware techniques for buffer insertion have also been proposed in [36, 45]. However, the approaches, in [36, 45], were based on traditional net-based techniques propagating continuous distributions and hence can produce over-buffered non-optimal solutions [23]. Here, we propose a new fuzzy optimization approach for variation aware simultaneous buffer insertion and driver sizing using a network-based algorithm. The deterministic version of the proposed BIDS algorithm is formulated to minimize the cumulative sum of buffers inserted and gate sizes. Delay constraints are modeled using required time variables at each node using a node based formulation as in, [87]. The delay constraints for buffer insertion and driver sizing is modeled as a piece-wise linear function of the buffer/driver types in the library. The optimization engine inserts zero or more buffers and increases driver sizes in each interconnect, considering the impact on circuit's critical path delay and the resource (buffer, driver/gate) cost efficiency.

Secondly, the previous works in buffer insertion are all performed after placement stage, where only incremental changes are possible. With increasing circuit complexity, it is becoming necessary to perform variation aware optimization early in the design phase. In this chapter, we propose the use of process variation aware circuit-wise buffer insertion and driver sizing formulation at the logic level. Most importantly, buffer insertion at the logic level requires careful abstraction of wire length, which is only available at the post placement stage. Here, we adopted the use of an accurate and fast interconnect length prediction technique at the logic level taking into account the number of cells/interconnects and fan-out of each cell. The technique is look-up table based wire length prediction and is similar to the one proposed in [38]. Further, solutions obtained from optimizations at the logic level can also be used as an estimate for planning during the layout level optimizations. The fuzzy optimization approach, as a pre-processing step, initially performs deterministic optimization with the variation parameters set to the worst case and average case values. The change in delay, due to buffer insertion/driver sizing is assumed to vary in between average and worst case value. The interval based delay variation is modeled as a triangular fuzzy number with a linear membership function. The results of these pre-processing deterministic optimizations are used to convert the uncertain fuzzy problem into a crisp nonlinear problem using the symmetric relaxation method [28, 56]. In the context of BIDS,

the crisp problem aims to maximize variation resistance (λ) or yield with circuit delay, power (buffer, driver cost) as constraints.

The proposed approach was tested on ITC'99 benchmark circuits and results indicate sizeable savings in buffer cost and driver sizes compared to the deterministic worst case approach. Finally, we also present a comparison of our logic level buffer insertion technique with a more accurate post layout version of the buffer insertion problem. The comparison is to highlight the efficacy of the wire length prediction mechanism. The difference in results on ITC benchmarks indicate that the logic level solutions are within 10% of the post layout level buffer insertion. The rest of the chapter is organized as follows. The problem formulation and the proposed fuzzy-BIDS framework is given in Sections 5.4. In Section 5.5, we present the experimental results followed by some conclusions in Section 5.6.

5.2 Modeling Delay Variations

In this work, we model the change in delay due to buffer insertion or gate sizing as an uncertain variable. The uncertain change in delay is modeled as a fuzzy triangular triple of the form Delay = ($Delay^m$, $Delay^l$, $Delay^u$). Here $Delay^m$ is the most possible value or the average case value and $Delay^l$ & $Delay^u$ corresponds to the lower and upper bounds, denoting the best and worst case changes of circuit delay. In accordance with previous works on process variations [36, 45, 87], we have also used the 3σ variation value for the worst case setting. The fuzzy programming problem similar to stochastic chance constrained programming framework, involves a relaxation step to convert the uncertain (fuzzy) constraints or objectives into a crisp (deterministic) framework. The relaxation in fuzzy programming starts with a set of deterministic optimizations by assuming the interval valued coefficients set to the worst and the average case setting. The results of these optimizations ($Result_{average}$ and $Result_{worst}$ and a variation resistance parameter (λ) is used to convert the fuzzy problem into a crisp problem. A brief outline of the fuzzy methodology for uncertainty-aware optimization is shown in Figure 5.1. Traditionally, the uncertainties due to process variations are usually handled using probability distributions. However, the probabilistic way of evaluating and optimizing the uncertainties is computationally expensive due to the need for complicated integration or large number of scenarios. Secondly, Buckley in [40], has shown that fuzzy programming based optimization guarantees solutions that are better or at least as good as their stochastic counterparts. The authors compared the

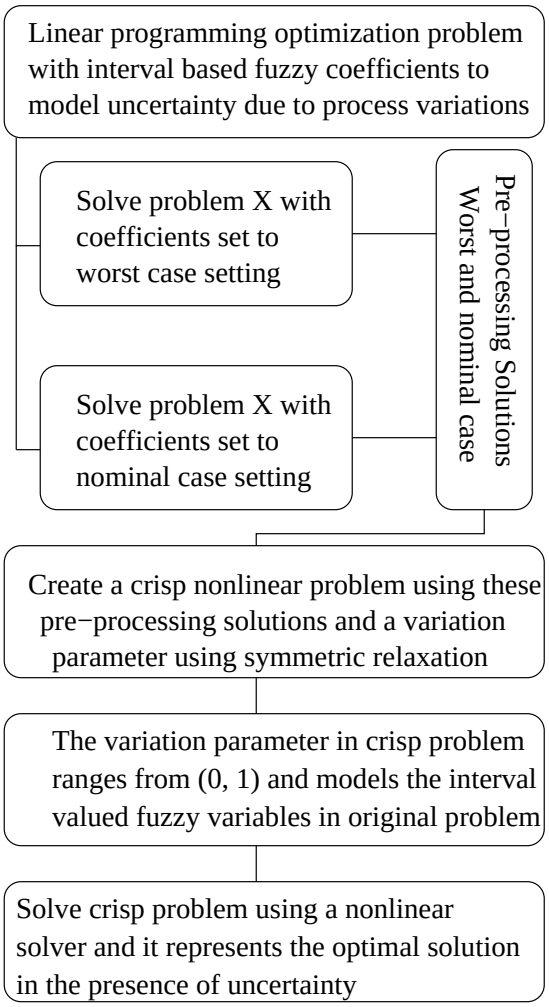


Figure 5.1 Fuzzy Programming Approach for Variation Aware Optimization

stochastic and fuzzy programming methodologies using Monte-Carlo simulations. The main difference between the techniques is that, the fuzzy optimization, in uncertain environments, finds the best solution (supremum operation over all feasible solutions) as opposed to averaging (integrals over all feasible solutions) in stochastic programming based optimization. Hence, fuzzy programming selects a solution which is better than or at least as good as the stochastic solution. The above arguments led us to investigate fuzzy programming approach to model uncertainty due to process variations in post layout and logic level buffer insertion and driver sizing problem.

5.3 Problem Formulation

In a placed and routed combinational circuit, after timing analysis, certain paths may violate the timing constraint. At this level, buffer insertion and driver sizing (BIDS) techniques have been able to successfully improve performance with a good power and noise trade-off. Without loss of generality, the list of standard cells (combinational gates/drivers) and interconnects in between register stages are considered for optimization. The BIDS technique can also be handled at the logic level with proper approximation of interconnect length. In the following sections, we explain the layout and logic level formulation of the buffer insertion and driver sizing problem.

5.3.1 Layout Level Modeling

Initially, drivers are fixed to minimum sizes and a fixed number of size increments for each driver are assumed to be available. The layout of the circuit is divided into n-regions and the density (number of devices to white space) of each region is calculated. The maximum size for each gate is decided based on the density of the region in which it is placed. The above restriction modeled as a bound for the variable in the optimization formulation. Similarly, for a set of candidate buffer locations, constraints are formulated for different buffer types with its associated output resistance, input capacitance and intrinsic delay. The layout level optimization for identifying candidate buffer locations were based on the concepts in [16]. The candidate buffer locations in this context, refers to the possible channels in critical interconnects where layout level optimization can insert buffers. The routed wires were divided into channels and channels in sparse regions were preferred as candidate buffer locations than denser ones. In addition to the density, channels which equally divide the critical connection are also preferred. The change in delay coefficients can be used to model the candidate buffer locations for each interconnect. The BIDS optimization problem aims to minimize resource cost, namely number of buffers inserted and total driver sizes increased , such that required arrival time at each primary output is less than the specified timing constraint.

The delay of a driver can be modeled as a linear function of its size and load from fan-out gates.

$$dg_i = a_i - b_i s_i + c_i * C_{load-j} \tag{5.1}$$

where, s_i refers to the size of driver i, C_{load-j} is the load seen from the driver, which is a function of the sizes of fan-out gates, constant coefficients a_i, b_i, c_i are empirically determined by extensive SPICE simulations for each gate in the library for various sizes and fan-out counts. Sizing gate-i improves the delay of current gate (as s_i increases) and increases the C_{load-j} seen by its fan-in gates. The interconnect delay on the other hand, has a quadratic proportionality to the length of the wire and is given by,

$$dint_i = R_0 * len_i(0.5 * C_0 * len_i + C_{pin}) \tag{5.2}$$

where, R_0 and C_0 refers to unit resistance and capacitance, C_{pin} the pin capacitance and len_i is the interconnect length. Buffer insertion on a interconnect impacts both the source gate delay (as a function of C_{load-j}) and interconnect delay (as a function of len_i). Since, the interconnect delay is a quadratic function of its length which can change during buffer insertion, we model the optimization problem using required arrival times and piece-wise arrival time changes during buffer insertion and driver sizing. The linear programming formulation is explained comprehensively in Section IV.

In nanometer regime, the wiring density has increased considerably leading to high aspect ratios in metal lines. This results in increased coupling between nets and can affect the timing and functionality of the circuits. Hence, in addition to considering process variations it is also necessary to minimize the effects of interconnect coupling to reduce losses due to timing yield failures. Noise on a net can be easily controlled during driver sizing. Interconnect coupling noise depends on the size of the driving gate (victim) and adjacently placed aggressor gates. Increasing the size of a gate increases the signal strength on the driven net and thereby the coupling noise on its victims. Hence, the up-sizing of a gate can increase the noise on the coupled nets and down-sizing a gate can reduce the same effect. Hence, we add a noise constraint to maintain the sizes of victim and the aggressor gates as in [25]. Secondly, the uncertainty due to process variations can be modeled as,

$$D = d_i + \sum_{j=1}^{n} d_j X_j + d_r X_r \tag{5.3}$$

where, d_i is the nominal delay and X_j and X_r are the random parameters representing correlated and independent variations respectively. The magnitude of these variations is given by the variables d_j and d_r, which is determined from extensive simulations. We capture these variations using the concept of

fuzzy numbers. The gate's delay is now a triangular value (average, low, high), instead of a single discrete value. Next, we explain the proposed fuzzy gate sizing approach for optimization in presence of process variations.

5.3.2 Logic Level Modeling

In the context of logic level modeling, the delay of a driver is again a linear function of the driver size and sum of the sizes of its fan-out gates. Hence, there is no significant difference between modeling the gate delay at the logic and the layout level. The interconnect delay on the other hand, has a quadratic proportionality to the length of the wire and is given by,

$$dint_i = R_0 * len_i(0.5 * C_0 * len_i + C_{pin}) \tag{5.4}$$

where, R_0 and C_0 refers to unit resistance and capacitance, C_{pin} the pin capacitance and len_i is the interconnect length. Accurate modeling of the interconnect length at the logic level is crucial to the effectiveness of the methodology. In this work, the wire length is obtained using a fast and accurate look-up table based estimation. Several researchers have worked on the problem of apriori length estimation. The authors in [94], have used the Rent's rule to derive the upper bounds for interconnection lengths of linear and square interconnection components. However, the rent's rule does not hold true at all levels of partition hierarchy in the nanometer era [38]. In this work, we use a look-up table based methodology taking into account the number of cells/interconnects and fan-out count of each cell. The estimation starts with the layout synthesis of a set of benchmark circuits. The benchmarks were selected from the MCNC benchmark suite and the complexity, in function of the number of gates ranges from 500 to 10000 approximately.

The layouts have to be generated for the target technology library. For each net a report is generated displaying its length and the fan-out count of its driver. Nets with same number of fan-out counts are grouped and the average net length for each fan-out count size is calculated. The table is then grouped based on benchmark circuit size and then averaged again. Hence, the array based table lookup requires circuit size (number of gates/nets) and fan-out count to extract the length of a interconnect at the logic level. The table is created with a maximum fan-out size of 20 and all interconnects with more than 20 fan-out gates are modularized to 20, before accessing the table. Buffer insertion on a

interconnect impacts both the source gate delay (as a function of C_{load-j}) and interconnect delay (as a function of len_i). The candidate buffer locations are selected with the objective of the diving the critical interconnect connections into equal halves. Since, the interconnect delay is a quadratic function of its length which can change during buffer insertion, we model the optimization problem using required arrival times and piece-wise arrival time changes during buffer insertion and driver sizing. The linear programming formulation, in the layout level context, is explained comprehensively in Section 5.4. The logic level formulation does not have the layout level constraints due to routing issues and the interconnect length is approximated using values from a look-up table.

5.4 Proposed Approach

In this section, we explain our modeling and solution methodology of the fuzzy buffer insertion and driver sizing (BIDS) problem in the presence of uncertainty due to process variations. Several formulations have appeared for the gate sizing and buffer insertion problems at various levels in the design flow. In this section, we describe a continuous linear programming approach to minimize resource cost with delay and noise constraints in the presence of variations.

5.4.1 Deterministic-BIDS

In this formulation, we start by explaining the modeling the delay constraints of gates and interconnect in the linear program. The interconnect delay is a quadratic function of the interconnect length and is significantly affected during buffer insertion. We formulate the delay constraints of the linear program using required arrival time and change in delay due to BIDS for each node from primary input to primary output. The required arrival time and improvement in delay based formulation enables the use of a piece-wise linear formulation for the nonlinear BIDS problem. The required time value of each sink node (req_j) must be greater than the sum of required time of its source node (req_i) plus gate (dg_i) and interconnect delay ($dint_i$) in between them. The slack in between the nodes can be improved by adding buffers $nbuf_i$ and sizing gates by a factor of $ngat_i$. Hence, the required time constraints can

be formulated as shown below,

$$req_i + dg_i + dint_i - Cb1_i * nbuf_i \quad (5.5)$$
$$-Cg1_i * ngat_i + Cg2_i * nfgat_i \leq req_j$$

$$req_i - Cb2_i * (nbuf_i - 1) - Cg1_i * ngat_i + \quad (5.6)$$
$$(dg_i + dint_i - Cb1_i) + Cg2_i * nfgat_i \leq req_j$$

where, $Cb1_i$ is the change in the delay due to the insertion of the first buffer and $Cb2_i$ is the change in the delay due to insertion of the subsequent buffers. The piece-wise required time constraints are inserted for all (i, j) driver-receiver pairs. The magnitude of the coefficients are also adjusted in accordance with the routing constraints and the candidate buffer locations. Inserting a second and third buffers tend to affect the delay lesser compared to the insertion of the first buffer. Hence, we use piece-wise require time formulation to model the change in delay due to buffer insertion. The $Cg1_i$ term is the change in delay due to gate sizing increments $ngat_i$ and is proportional to the coefficient b_i in gate delay (Equation. 5.1). The term $nfgat_i$ is the change in the sizes of the fan-out gates of node i and its coefficient $Cg2_i$ is proportional to the coefficient c_i in Equation 5.1. The $Cb1_i$ and $Cb2_i$ coefficients also depends on the candidate buffer locations of each interconnect i. The node based required arrival time formulation also avoids the exponential complexity of the path based formulation. The required time constraints are then related to the final timing objective as $req_i <= T_{spec} \quad \forall i \epsilon PO$. In addition to the delay constraints, the impact of coupling capacitance on timing of the circuit is also modeled. A pre-processing step identifies, the set of aggressors and a constraint $ngat_{victim-size} - ngat_{aggressor-size} <= 1$ is added for each victim gate. The deterministic version of the buffer insertion and driver sizing

problem can be shown as,

$$min \sum_i (nbuf_i + ngat_i) \tag{5.7}$$

$$s.t. \quad req_i \leq T_{spec}; \quad \forall i \varepsilon PO;$$

$$s.t. \quad req_i + dg_i + dint_i - Cb1_i * nbuf_i$$

$$-Cg1_i * ngat_i + Cg2_i * nfgat_i \leq req_j$$

$$s.t. \quad req_i - Cb2_i * (nbuf_i - 1) - Cg1_i * ngat_i +$$

$$(dg_i + dint_i - Cb1_i) + Cg2_i * nfgat_i \leq req_j$$

$$s.t. \quad ngat_{victim-size} - ngat_{aggressor-size} <= 1;$$

The coefficients $Cg1, Cb1$ and $Cb2$ in this formulation, which is assumed to be varying between worst case and best case bound, are modeled using fuzzy numbers with linear membership function. The fuzzy modeling and optimization methodology is explained next.

5.4.2 Fuzzy-BIDS

Fuzzy optimization techniques provide an efficient mechanism for modeling and optimizing systems that exhibit imprecision and variations. The fuzzy mechanism starts with a set of pre-processing optimization with the varying parameters set to worst case and nominal case values. In this work, we model uncertainty due to process variations, as an imprecision in the delay improvement due to BIDS. The coefficients $Cg1, Cb1$ and $Cb2$, which control the improvement in delay due to buffer insertion and driver sizing are modeled as triangular fuzzy numbers. The worst case values of these coefficients are assumed to be $Cg1 - Vg1, Cb2 - Vb2, Cb1 - Vb1$, where the values $Vg1, Vb2, Vb1$ are selected to create a worst case (3σ) delay variation, in accordance with recent variation aware optimization frameworks [35, 36, 87].

The optimization problem in Equation 5.7 is solved with the worst case coefficient setting $Cg1 - Vg1, Cb2 - Vb2, Cb1 - Vb1$ and the results of this optimization is referred to as Obj_{wc}. Similarly, the optimization problem in Equation 5.7 is solved with the nominal values and the results are referred to as Obj_{nc}. The results Obj_{wc}, Obj_{nc} and a new variation parameter λ are used to transform the uncertain fuzzy optimization problem into a crisp nonlinear programming problem using the symmetric

relaxation method [28,56]. The crisp nonlinear problem for BIDS in the presence of process variations is given by the following equation.

$$
\begin{aligned}
&maximize \ \ \lambda \qquad (5.8)\\
&s.t.\ \lambda(Obj_{nc} - Obj_{wc}) -\\
&\sum_i (nbuf_i + ngat_i) + Obj_{wc} \leq 0;\\
&s.t.\ req_{po} \leq T_{spec}; \quad \forall p \varepsilon PO;\\
&s.t.\ req_i + dg_i + dint_i - (Cb1_i - Vb1_i * \lambda) * nbuf_i\\
&-(Cg1_i - Vg1_i * \lambda) * ngat_i + Cg2_i * nfgat_i \leq req_j\\
&s.t.\ req_i - (Cb2_i - Vb2_i * \lambda) * (nbuf_i - 1) -\\
&(Cg1_i - Vg1_i * \lambda) * ngat_i + (dg_i + dint_i -\\
&(Cb1_i - Vb1_i * \lambda)) + Cg2_i * nfgat_i \leq req_j\\
&s.t.\ ngat_{victim-size} - ngat_{aggressor-size} <= 1;
\end{aligned}
$$

where, the parameter λ is bounded by 0 and 1. The parameter λ can take any values between 0 and 1, for the BIDS problem. The crisp optimization problem has four variables, delay, noise, resource cost and process variations (λ) in the above formulation. The parameter λ, as seen from the formulation, maximizes variation resistance and also controls the delay and resource cost values in the optimization. Hence, a maximization of the variation resistance parameter simultaneously guarantees, high yield, low cost and low delay. It has been shown for problems in other application domains that the above formulation provides the most satisfying optimization solution in the presence of uncertainty [56].

5.5 Simulation Methodology and Results

The proposed fuzzy linear programming optimization for buffer insertion and driver sizing was tested on ITC'99 benchmark circuits. First, the RTL level VHDL netlists are converted to a flattened gate level Verilog netlist using the synopsys design compiler. Verilog file from the design compiler is then placed and routed using the cadence encounter. The TSMC db, lef and tlf libraries are used to synthesize the benchmark circuits. The placed and routed netlist (DEF file), library of cell delay

information and the Verilog file are given as an input to a C script (*DEF2AMPL*), which converts the netlist into a AMPL based mathematical program format. The DEF2AMPL script is then used to generate the linear programming models for the benchmarks with delay coefficients set to mean and the maximum possible variation (worst case). The change in delay coefficients due to buffer insertion and driver sizing is chosen in accordance with the gate and interconnect delay variation values in, [12, 79]. The *DEF2AMPL* script uses the results of these optimizations and generates a crisp nonlinear AMPL model. The crisp nonlinear optimization problem is solved using the KNITRO solver to find the optimal gate sizes in presence of variations in gate delay.

The logic level simulations are performed without using the place and route tools. Depending on the target technology library, a look-up table is created for predicting wire length with the number of gates of the circuit and the number of fan-out count as the parameters. The look-up table is built from extracting wire length values from previously placed and routed benchmarks. The predicted wire length values, library of cell delay information and the logic level Verilog netlist are given as an input to the same C script (*DEF2AMPL*), which converts the netlist into a AMPL based mathematical program format. The DEF2AMPL model first creates a worst case model with timing as the only objective, to identify the best performance in which the circuit can operate. The timing value identified in this step is used as a constraint in the following simulations to optimize the cost (buffers and driver sizes) with nominal case, worst case and fuzzy modeling of variations. The DEF2AMPL script is then used to generate the piece-wise linear programming models for the benchmarks with delay coefficients set to mean and the maximum possible variation (worst case). The complete simulation flow for the logic level experiment is shown in Figure 5.2. The layout level simulation approach bypasses the wire-length prediction step, as actual interconnect length values are available after place and route. The cadence encounter place and route tools are used on the gate level netlist to generate the initial layout.

5.5.1 Layout Level BIDS

The buffer and gate resource cost reduction achieved by the fuzzy sizing approach compared to worst case deterministic sizing is documented in Table 5.2. The table also provides information on the circuit characteristics and the complexity (constraints) of the fuzzy optimization formulation. The worst case BIDS results in column 5, correspond to the delay coefficients set to their maximum varia-

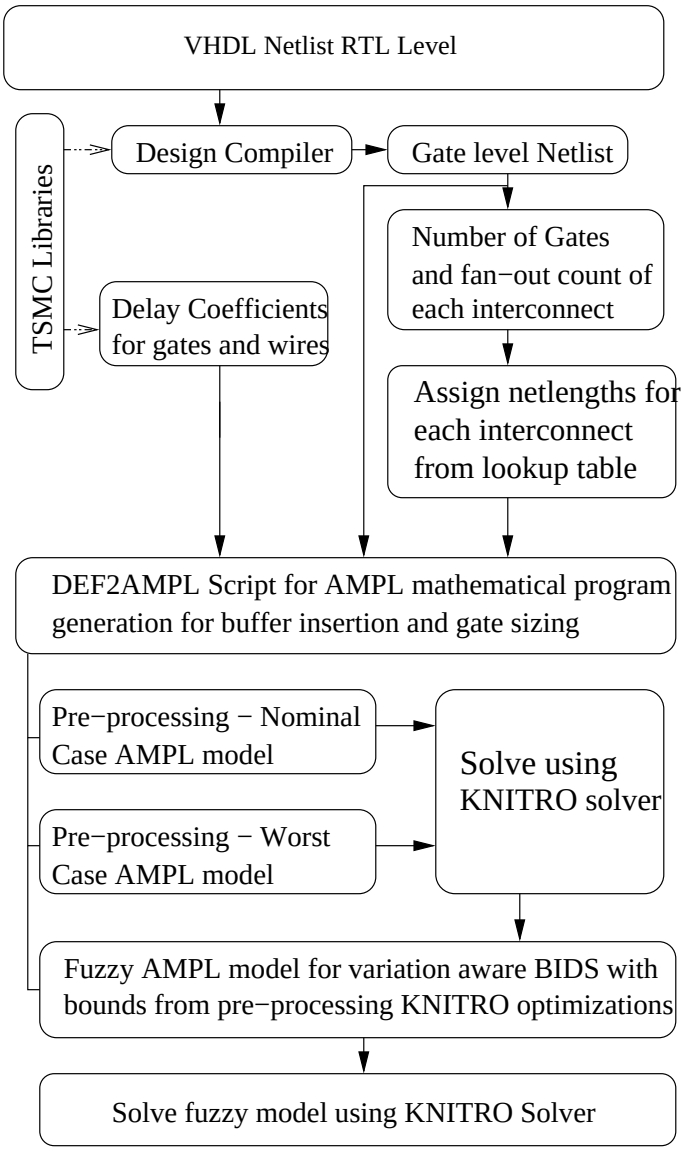


Figure 5.2 Simulation Flow - Fuzzy BIDS

tion case for high yield. It can be seen that there is a sizable savings in buffer and gate cost by using the fuzzy sizing approach as compared to deterministic worst case gate sizing approach. The execution time of the fuzzy optimization approach is also shown in Table 5.2. The continuous solutions from the proposed approach were rounded to get discrete buffer numbers and gate sizes, with very little effect on timing and resource cost.

Table 5.1 Layout Level Results on Benchmark Circuits

ITC' 99 Benchmark	No. of of gates	No. of Nets	No. of Constraint	Buffer & Gate Cost		Objective Value (λ)	Percent Change	Runtime (secs)
				DWC	Fuzzy			
b13	249	269	1027	19	14	0.55	26.4%	0.97
b11	385	322	1516	70	46	0.57	34.2%	1.85
b12	834	847	3810	100	80	0.58	20%	70.2
b14	4232	4544	30437	830	330	0.69	60.21%	200
b15	4585	4716	31951	591	260	0.74	56.00%	1185
b20	8900	9538	63855	991	408	0.68	58.8%	3800
b22	12128	13093	87108	534	237	0.66	55.6%	5791
Average Percent							44.5%	

Legend - DWC: Deterministic Worst Case BIDS Approach; Fuzzy: Fuzzy BIDS Formulation;
Legend - Constraints: No. of Constraints in FLP program; Percent Improvement: Fuzzy Vs DWC;

5.5.2 Logic Level BIDS

The buffer and gate resource cost reduction achieved by the fuzzy sizing approach compared to worst case deterministic sizing is documented in Table 5.2. The table also provides information on the circuit characteristics and the complexity (constraints) of the fuzzy optimization formulation. The worst case BIDS results in column 5, correspond to the delay coefficients set to their maximum variation case for high yield. It can be seen that there is a sizable savings (35% on the average) in buffer and gate cost by using the fuzzy sizing approach as compared to deterministic worst case gate sizing approach. The execution time of the fuzzy optimization approach is also shown in Table 5.2. The continuous solutions from the proposed approach were rounded to get discrete buffer numbers and gate sizes, with very little effect on timing and resource cost. Also, without loss of generality, the proposed methodology is assumed to optimize gates/interconnects in between two register stages. With increasing performance targets, the number of levels of logic inbetween register stages are decreasing. Hence, the number of gates in between two register stages is reducing. Thus, we feel that optimizing and reporting results on upto 12000 gates is a good indicator on large industrial benchmarks.

Figure 5.3 compares the results in number of buffers and gate sizes reported by the logic level and the layout level simulation. It can be clearly seen, that the results are very close between the logic and layout level formulations for all the benchmarks, with b20 being an exception. However, even with b20 included, the logic level optimization on the average is within 10% of the placed and routed buffer insertion and driver sizing results, indicating the efficacy of the wire length prediction and logic level buffer insertion mechanism.

Table 5.2 Logic Level Results on Benchmark Circuits

ITC' 99 Benchmark	No. of of gates	No. of Nets	No. of Constraint	Buffer & Gate Cost		Percent Improvement	Runtime (secs)
				DWC	Fuzzy		
b13	249	269	1027	33	20	39.3%	4
b11	385	322	1516	67	47	29.8%	2.5
b12	834	847	3810	86	64	25.5%	65
b14	4232	4544	30437	425	309	27.4%	190
b15	4585	4716	31951	555	332	40.3%	1025
b20	8900	9538	63855	1014	585	41.1%	3500
b22	12128	13093	87108	550	298	45.6%	5650
Average Percent						35%	
Legend - DWC: Deterministic Worst Case BIDS Approach; Fuzzy: Fuzzy BIDS Formulation; Legend - Constraints: No. of Constraints in FLP program; Percent Improvement: Fuzzy Vs DWC;							

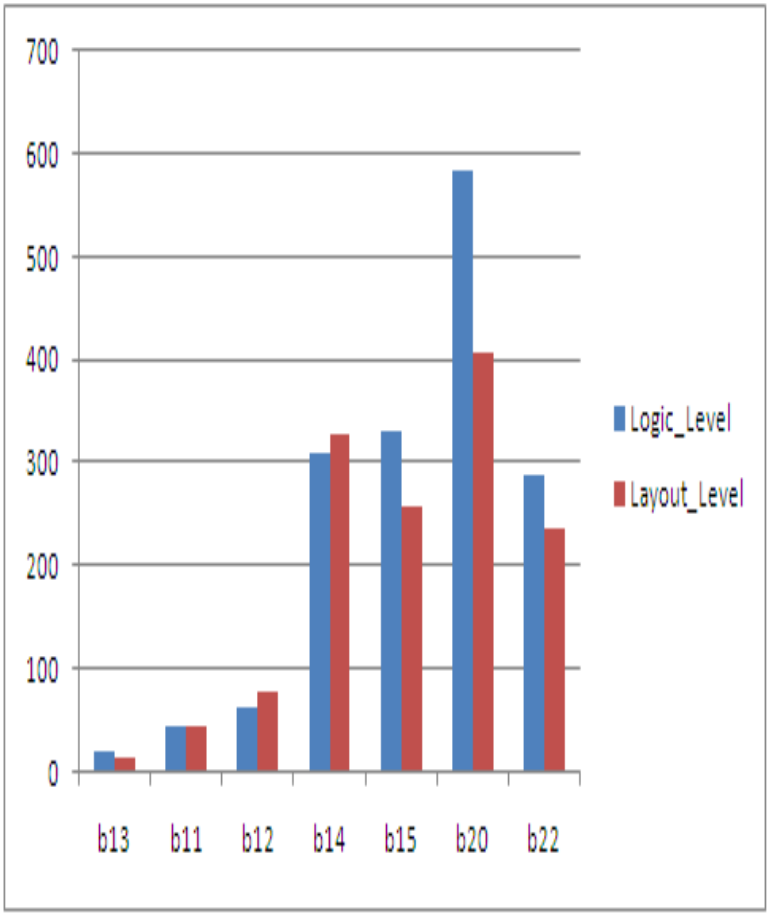


Figure 5.3 Fuzzy BIDS: Logic Level Versus Layout Level

5.6 Conclusion

In this chapter, we proposed a new approach for simultaneous buffer insertion and driver sizing at the logic level considering process variations using fuzzy linear programming technique. An ac-

curate and fast interconnect length prediction technique was developed at the logic level taking into account the number of cells/interconnects and fan-out of each cell. The proposed fuzzy BIDS approach maximizes variation resistance (robustness) of the circuit, with delay as constraint and cost bounded between the worst and nominal case values. Experimental results on ITC'99 benchmark circuits indicate sizable savings in resource cost compared to the deterministic worst case approach. The logic level buffer optimization technique was also compared with post layout buffer insertion technique, for accuracy of wire length prediction, and the difference in results is found to be less than 10%.

CHAPTER 6

DYNAMIC CLOCK STRETCHING

6.1 Introduction

The power-performance trade-off in the nanometer era, has only exacerbated with the inception of parameter variations in nanometer technology. Parameter variations comprise process deviation due to doping concentration, temperature fluctuations, power supply voltage variations and noise due to coupling. Variations can cause frequency and power dissipated to vary from the specified target and hence can result in parametric yield loss. Parametric yield, in this context, is defined as a design's sensitivity to variations, and is expected to cause 60-70% of all yield losses in the impending technology generations [72]. To ensure correct operation under all possible variations (process, voltage and temperature), circuits are often designed with a conservative margin. The margins are added to the voltage and/or device structures to account for the uncertainty due to worst case combination of variations. However, such a worst case combination is very rare or even impossible in most situations making this design strategy overly conservative.

In this context, several researchers have proposed the use of statistical timing analysis and statistical optimization mechanism to meet timing in the presence of variations without significant over design [11, 49, 74, 84, 87]. The variation aware optimization methodologies use stochastic or fuzzy methodology to minimize the impact of uncertainty due to process variations on performance, power and other design overheads. Statistical timing analysis (SSTA) was investigated in, [11, 74], where continuous distributions are propagated instead of deterministic values to find closed form expressions for performance in presence of variations. Variation aware solutions have also been developed for circuit optimization problems like gate sizing, buffer insertion and incremental placement [49, 84, 87]. The main objective of these works has been to improve yield, without compromising on performance, power and area. The variation aware optimization techniques have shown to improve design overheads without loss in parametric yield. However, the statistical optimization methods still over consume re-

sources irrespective of whether the circuit is affected by variations or not. Hence, to facilitate more aggressive power-performance-yield tradeoff improvement, dynamic schemes to detect and correct the uncertainty due to process variations are becoming necessary. Further, the authors in [76], proposed a novel design paradigm which achieves robustness with respect to timing failure by using the concept of critical path isolation. The methodology isolates critical paths by making them predictable and rare under parametric variations. The top critical paths, which can fail in single cycle operation, are predicted ahead of time and are avoided by providing two cycle operations. The methodology works well for special circuits with rare critical paths, however has severe timing penalty on benchmark designs.

One of the popular methods to dynamically combat process variation's impact on design has been to use adaptive voltage scaling (AVS) [24,48,75]. The voltage scaling systems tracks the actual silicon behavior with an on-chip detection circuit and scales voltage in small increments to meet performance without high overheads in the presence of process variations. In [3,82], the critical path of the system was duplicated to form a ring oscillator and actual performance requirement of the circuit is co-related to the speed of the oscillator and appropriate voltage scaling is performed. However, in the nanometer era, it is not feasible to use a single reference for a critical path and the variations spread, can make the close to critical delay paths critical on actual implementation. Recently the authors in [48], proposed an AVS system which can emulate critical paths with different characteristics. With increasing amount of on-chip variations and spatial correlation the methodology can have severe discrepancies. In a bid to reduce such margin and remove the dependency of feedback mechanism on a single path, a novel on-chip timing checker was proposed in [24] to test a set of potential critical paths. The method uses a shadow latch with a delayed clock to capture data in all potential critical paths. An error signal is generated if the value in original and shadow latch is different due to a timing violation caused by process variations. The methodology however aims at correcting (not preventing) errors caused by aggressive dynamic voltage scaling. To guarantee high timing yield and low overheads in the presence of variations, the ultimate solution is to dynamically alter the clock signal frequency.

The authors in [33,34], proposed a technique to control and adjust clock phase dynamically in the presence of variations. The methodology focused on the design of a dynamic delay buffer cell that senses voltage and temperature variations and alters clock phase proportionately. However, it is not generic to all types of variations and does not include spatial correlation between the delay buffer and the gates in the critical path. Plus, the methodology is not input data dependent and hence changes the

clock capture trigger in more than required number of instances. In this chapter, we propose a new approach for dynamic clock stretching by dynamically detecting delay due to process variations. Here, instead of modulating the clock duty cycle, we delay the capture clock edge to critical memory cells to accommodate the increased signal propagation delay due to variations. The methodology captures the signal transition halfway in the critical path in a positive level triggered latch. However, if the signal transition on the critical path, which is expected to occur before time T/2 (positive level of clock) is delayed due to process variation. The latch in the detection circuit holds an opposite value compared to the signal line and a delay-flag is set. Here, T is the clock cycle time. The delay-flag dynamically stretches the clock at the destination memory flop, to accommodate the extra signal propagation delay due to variations. Thus the clock stretching methodology avoids a mismatch in the data being captured and hence prevents timing error. The detection circuitry needs to be added to the top ”n” critical paths and an error signal from any of these paths can stretch the clock in the appropriate destination memory cell. The clock is stretched (the capture edge trigger is delayed) considering both spatial correlations between closely spaced critical path gates and an average variation range as reported, in [12, 79]. The proposed methodology is demonstrated on example critical paths to explain the functioning of methodology. Experimental results based on Monte-Carlo simulations on ITC’99 benchmark circuits indicate efficient improvement in timing yield with negligible area overhead. The rest of the chapter is organized as follows. In Section 2, we explain the construction of the delay detection circuit for efficient clock stretching . Experimental evaluation and results for example circuit configuration and benchmark circuits is presented in Sections 3 and 4 respectively. The chapter is concluded in Section 5.

6.2 Proposed Methodology

In this section, we explain the process variation aware delay detection circuitry and dynamic clock stretching methodology. The delay detection and clock stretching logic (CSL) is added only on critical and near critical path cells to accommodate the increased data path delay due to process variations. In the presence of delay due to variations in a path, the CSL circuit of that critical path delays the instant of the active clock edge trigger on the destination memory flops. Popular dynamic schemes like frequency scaling require some activation time (PLL delay) to adjust (increase/decrease) the clock

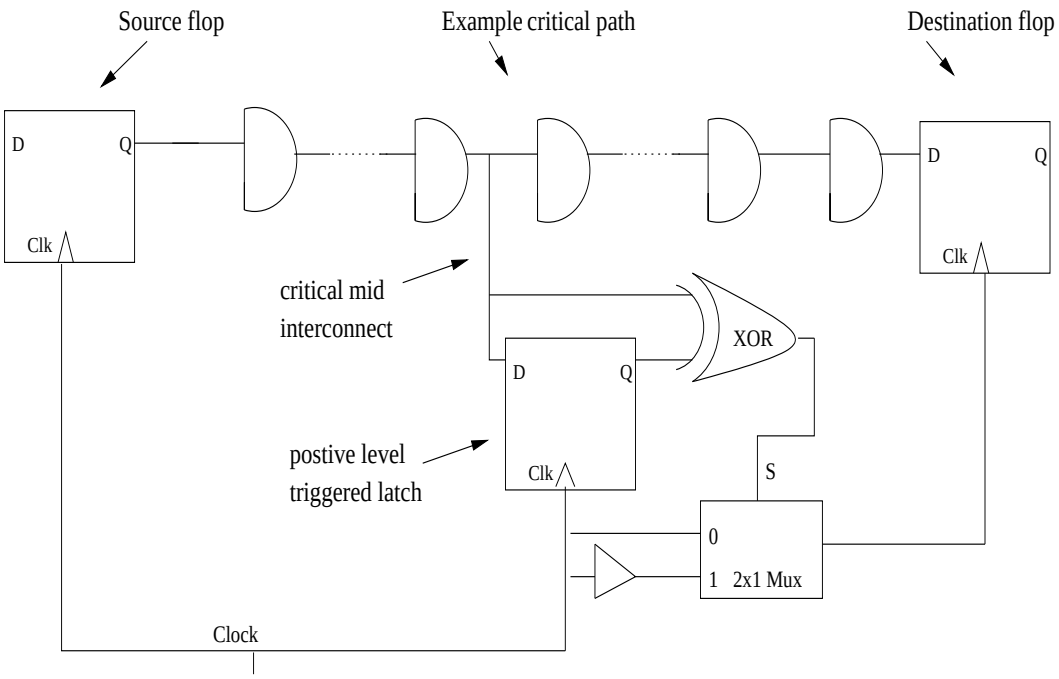


Figure 6.1 Dynamic Clock Stretching for Variation Tolerance

frequency. The proposed methodology can provide immediate activation and enable prevention of timing failures. The proposed CSL mechanism detects and stretches the clock in the same clock period. Further, since the detection circuit monitors data transitions on critical net, the methodology is independent of the type of process variation. Hence, it is suitable for tolerance against delay due to process, temperature, voltage variations and even coupling noise. The proposed dynamic detection and clock stretching technique is shown in Figure 6.1. A critical path in between flip-flop stages, the transition capture latch, delay-flag setting XOR gate, clock stretching buffer and a multiplexor is shown in Figure 6.1. The critical path operates normally in the absence of variations, where the data path propagation delay is within the clock cycle time (T). Any variation, due to process, voltage, temperature or coupling, in the critical path forces propagation delays to increase, which can lead to synchronization errors in destination flip-flop. The only solution, other than resorting to a conservative timing specification, is to dynamically stretch or delay the capture clock trigger in the destination flops. Hence, the clock stretching technique enables the capture of correct data in the presence of variations without significant over design.

The delayed transition detection and clock stretching circuit have to be provided for the entire top ”n” most delay critical paths. Since, any of these paths can become the most critical path and violate

the timing specification due to process variations. The paths with a delay value within 15% of the most critical path are selected as the candidates for dynamic stretching [12, 24]. Another important pre-processing step would be the identification of critical locations (interconnects), halfway in the critical path, where delay due to variations can be detected. In normal conditions, the transition on the critical interconnect should settle before the negative edge of the clock. In other words, the transition on the critical interconnect should happen, before the negative edge of the clock in the absence of variations. In the presence of variations, the transitions have to be after the negative edge of the clock due to the increased delay. We can use timing analysis results, to find a critical interconnect in each of the top ”n” critical paths of the circuit. If a critical interconnect does not exist automatically in any of the top ”n” critical paths. We can perform circuit sizing, buffer insertion or other incremental changes to create a net with the specified characteristics. Another important aspect in the proposed technique is checking the feasibility of capturing the delayed transition with reference to the negative edge of the clock. The key issue is to make sure the critical transition is not missed due to setup/hold timing violation of the positive level triggered latch. In an effort to confirm, we constructed an example critical path with 20 levels of logic at the 65nm technology node level. We used real delay information from 65 nm technology library for this experiment. The simulations were performed on critical paths with multiple logic gate configurations and a 15% range for delay changes due to process variations. The setup clearly confirmed the feasibility of the proposed transition capture methodology.

The positive level triggered latch, shown in Figure 6.1, captures the value floating on critical interconnect during the positive level of the clock. The critical transitions on this interconnect, on normal conditions, needs to have the transition completed before time (T/2). If the transition had been delayed due to process variations, then the inputs to the XOR gate will be different. Hence, in the presence of delay due to variations the XOR gate will output a 1. The multiplexor, shown in Figure 6.1, has the ability to select the normal (undelayed) or delayed clock for the destination flip-flop. The XOR gate output (referred to as the delay-flag in earlier sections) is used as the select line of the multiplexor. Thus, the proposed methodology dynamically selects the delayed clock, in case the signal propagation is delayed in the data path due to variations. An important assumption in the proposed methodology is the property of spatial correlations. Spatial correlations property defines, that closer a set of devices are placed, the higher is the probability that they have a similar variation range [37, 87]. Plus, in the nanometer era due to the increased timing requirement, gates in the top ”n” critical paths

Table 6.1 Description of Symbols in Simulation Snapshot

Clock signals	
CK	Circuit Clock
delclk	Delayed clock
muxoutclk	Output clock from multiplexor
Data signals	
in11	Input data value
a5	Critical mid interconnect value
out1	Output data value
Clock stretch signals	
a5test	Latch output
muxsel	Select line of multiplexor

are usually placed closer to each other. Hence, we can safely assume the irrespective of the gate's level in a critical path, if one gate is affected by variations, there is high probability other gates in the critical path are affected in similar fashion. For example, if there is a variation in the second half of the path, due to the property of spatial correlations some gates in the first half of the path will also affected by variations and vice versa. The magnitude of these variations, will be hard to predict, and can be different based on their location in the chip area. However, the presence or absence of variations can be safely assumed with the property of spatial correlations.

Further, the use of a delayed clock edge trigger for the destination flops, can result in timing inconsistency. The main issues in this context, are the short paths and consecutive critical paths. A short path raises the possibility of data corruption (failures) in a destination memory cell. However, in nanometer designs short paths are usually rare due to the multiple objectives of power, performance and yield. Plus, with a small margin of clock stretching (10-15%), it is easy to perform sizing to eliminate short path failures. Secondly in pipeline circuits if a critical path is followed by another critical path in the following pipeline stage, the CSL methodology can cause timing failures. This is because the delayed clock circuitry reduces the data capture time available in subsequent pipeline stage. Hence, in pipeline circuits with consecutive critical paths, the delay-flag has to be propagated to subsequent stages to create the necessary slack by automatic clock stretching. In the next section, we validate the proposed CSL technique on simple example and benchmark circuits.

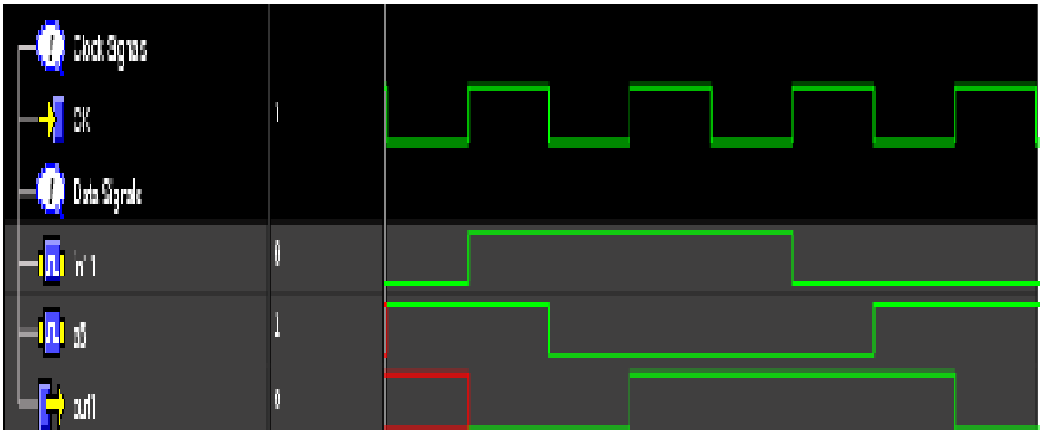


Figure 6.2 Simulation Snapshot of Example Circuit: No Variations; No Clock Stretching

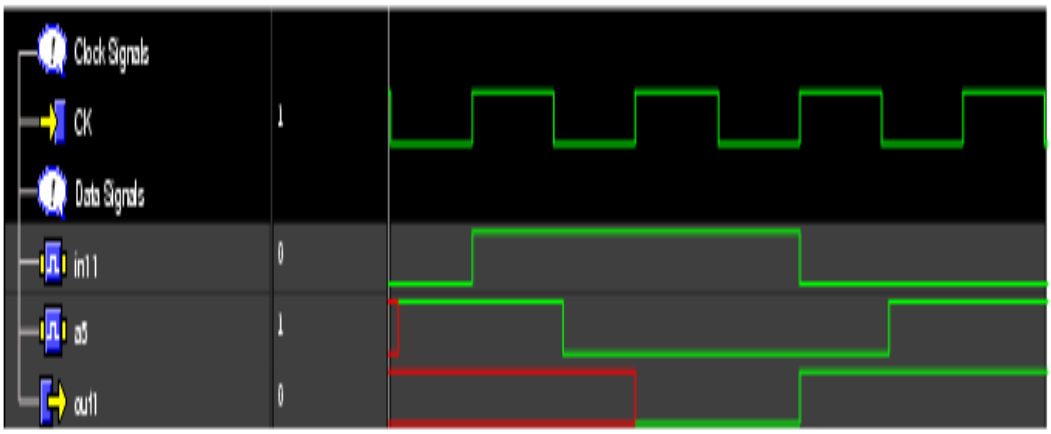


Figure 6.3 Simulation Snapshot of Example Circuit: With Variations; No Clock Stretching

6.3 Experimental Evaluation

To evaluate the proposed methodology, we simulated an example circuit using Cadence NCVerilog simulator. The purpose of this simulation is to elucidate the functionality of the methodology in the presence of variations in circuit elements. The efficiency of the methodology, however is calculated with Monte-Carlo based timing yield simulations. A chain of inverters in between two flip-flops stages is chosen as the example circuit. In this circuit, all interconnects in the path makes a transition. Hence, the net halfway in the path becomes the necessary critical interconnect. The clock cycle time is selected in reference to the critical path delay at the nominal corner. In other words, the clock cycle time is chosen without adding a margin for uncertainty in delay due to process variations. In addition to the input, output and clock signals, a critical interconnect that transitions from $0 \rightarrow 1$ (or $1 \rightarrow 0$) just

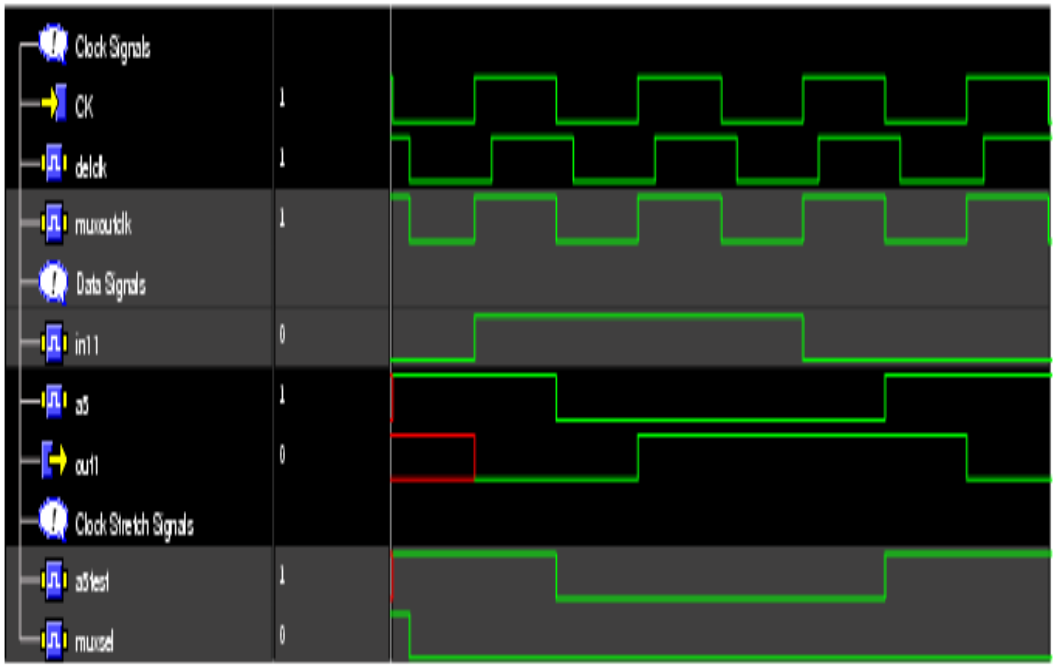


Figure 6.4 Simulation Snapshot of Example Circuit: No Variations; With Clock Stretching

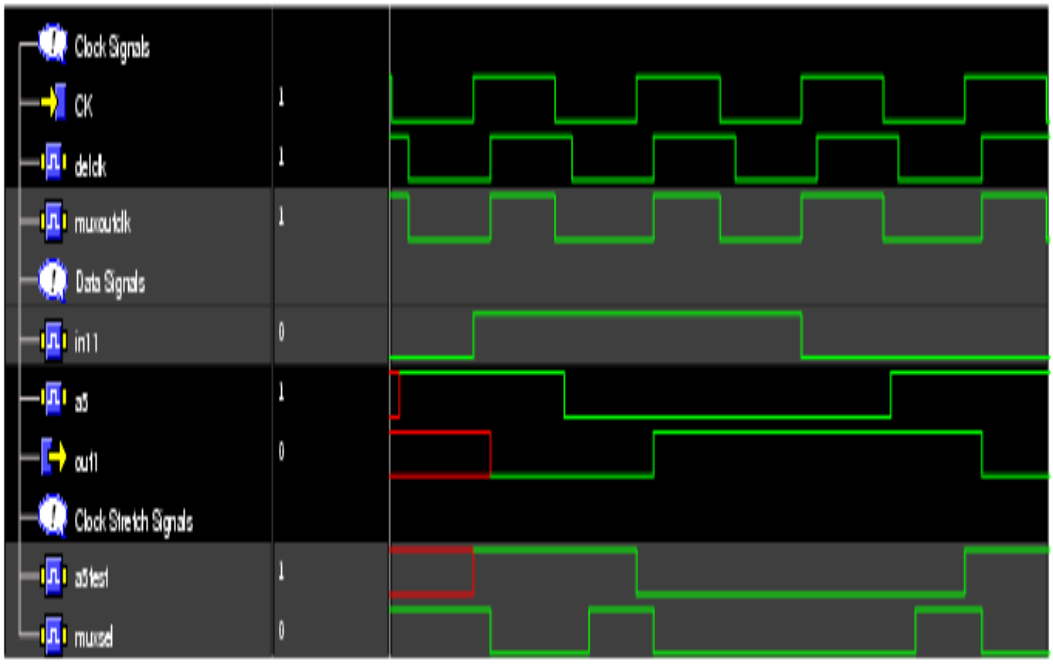


Figure 6.5 Simulation Snapshot of Example Circuit: With Variations and Clock Stretching

before the negative edge of the clock is also displayed in the simulation snapshots (Figures 6.2 and 6.3). A simulation snapshot of the example circuit with no variations is shown in Figure 6.2. The signals shown in the snapshot are classified as clock (CK) and data (in11, a5, out1)signals. The signal

in11 is the primary input (output from the source flip-flop), a5 is the critical interconnect halfway in the path and out1 is the output signal connected to the destination flip-flop. It can be seen from Figure 6.2, that a $0 \rightarrow 1$ transition on the input signal (in11), initiates a transition on a5 and is captured on the next clock cycle in the destination flip-flop (out1). The simulation snapshot for the same circuit in the presence of uncertainty in delay due to process variations is shown in Figure 6.3. In the presence of variations, the same $1 \rightarrow 0$ transition on critical net a5, happens after the negative edge of the clock. The delay due to process variations also caused a timing violation on the output flip-flop (out1), as the $(0 \rightarrow 1)$ transition is captured on the subsequent clock cycle.

Hence, the transition on the critical interconnect, which is an indicator of a timing violation is sent to the clock stretching logic (CSL) unit. The latch in the CSL unit captures the transition if it happens in the positive level of the clock. If the transition is delayed due to process variations, the inputs to the XOR gate (signal line and latch output) differ in value, causing a 1-output in the XOR gate. The output of the XOR gate, which is connected to the multiplexor, selects the delayed clock. The delayed clock sent to the destination flip-flop, avoids a timing violation. The simulation snapshot for the example circuit with the clock stretching logic is shown in Figures 6.4 and 6.5. The delayed clock (delclk) and the mux output to the destination flop (muxoutclk) are added to the list of clock signals. In addition to the input and clock signals, the CSL snapshots also show clock stretch signals a5test and muxsel. The muxsel signal is the output of the XOR gate and a5test is the output of the positive level triggered latch. It can be clearly seen in Figure 6.5, that the delayed clock is sent to the muxoutclk, whenever the transitions on the critical interconnect happens after the negative edge of the clock. Further in Figure 6.4, we show that in the absence of variations the circuit operates normally and clock (CK) is used at the source and the destination flip-flops. A brief description of the symbols used in the simulation snapshot is shown in Table 6.1. Similar to the simulation snapshot, the signals in the symbol table are also grouped as clock, data and clock stretch signals.

6.4 Simulation Results

The proposed dynamic clock stretching logic was tested on ITC'99 benchmark circuits. The improvement in timing yield for the circuits was estimated using Monte-Carlo simulations. The simulation flow for the timing yield estimation is shown in Figure 6.6. The gate and net delay of the

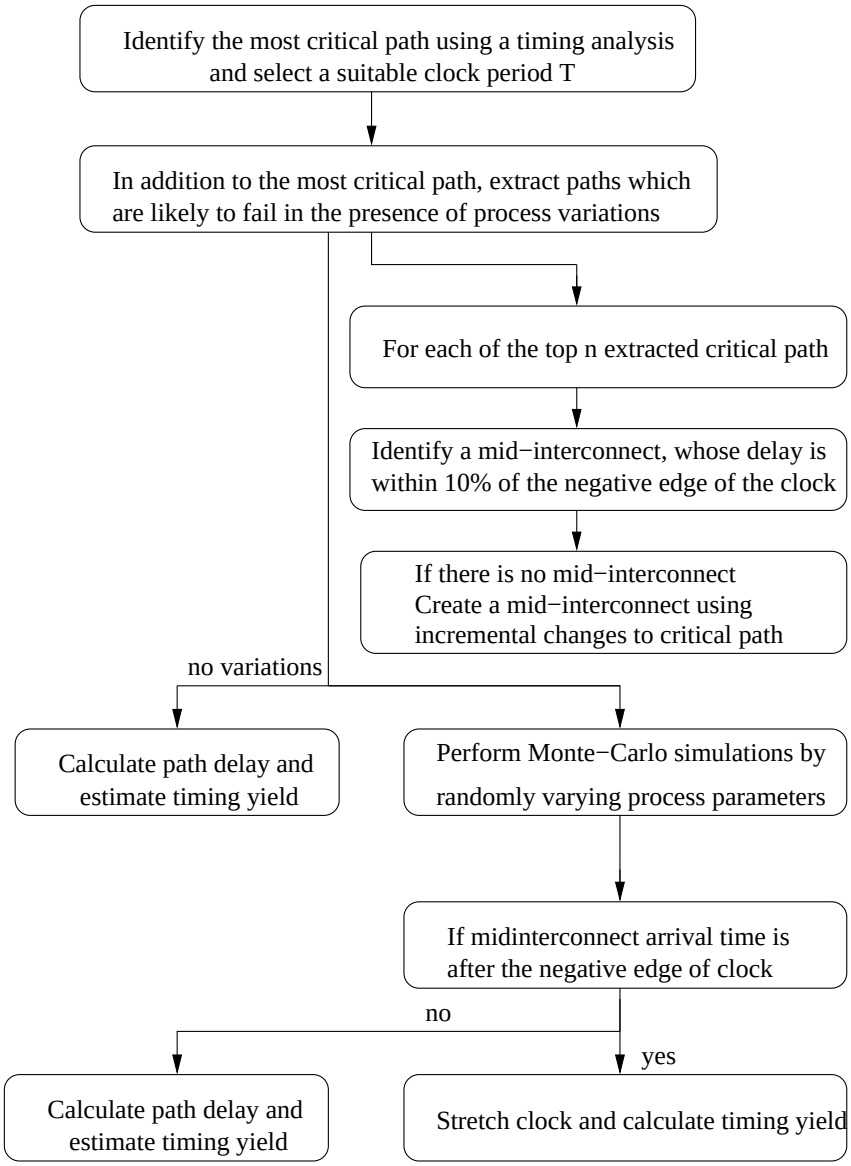


Figure 6.6 Simulation Flow for Timing Yield Estimation

circuit elements were assumed to have a variation range of around 20% from the nominal value. In the absence of real statistical data, it has been pointed out in [79], that it's reasonable to assume a variation parameter value of around 20-25% on the delay due to process variations. The RTL level VHDL netlists of the benchmark circuits were converted to a flattened gate level Verilog netlist using the Synopsys design compiler. The output Verilog file from the design compiler is then placed and routed using the cadence encounter tool. A timing analysis report (TARPT) file is then generated to identify the critical paths whose delay value is within 15% of the most critical path. A Monte-Carlo simulation framework is created in a C-program environment for the ITC benchmarks with placed

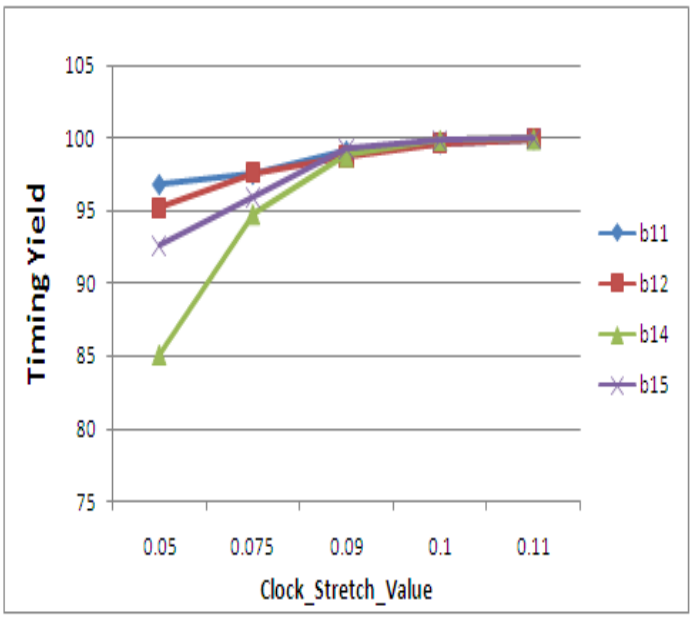


Figure 6.7 Clock Stretch Range Versus Timing Yield

and routed (DEF), parasitics file (SPEF), timing analysis report (TARPT) and standard cell delay libraries as input. The Monte-Carlo simulation creates 20000 instance of the benchmark with varied delay between nominal and maximum variation range to estimate the timing yield. Timing yield in this context, is defined as the percentage of circuit meeting the timing specification. The circuits are tested for timing yield in two configurations, namely (i) original circuit with variations and (ii) CSL added circuit with variations. The timing specification for the original and the CSL added circuit, is assume to be 100% in the absence of variations.

The improvement in timing yield achieved by the CSL methodology compared to the original configuration is shown in Table 6.2. It can be seen that the average timing yield of the original circuit with the nominal timing specification is approximately 60% and is not acceptable. Hence, circuit designers (with or without statistical optimization) close timing with an extra margin. The extra timing margin increases the overhead and/or decrease the performance at which the circuit can operate. The proposed methodology dynamically detects the delay due to variations and adds the extra timing margin only when required. The proposed CSL methodology has increased the average timing yield to around 99.9%. The clock was stretched to create an extra timing slack of 10% only if delay due to process variations are activated in the worst case critical paths. In the context of timing failures due to short paths, it is crucial to keep the clock stretching range as short as possible. Hence, we performed

Table 6.2 Timing Yield Results on Benchmark Circuits

ITC' 99 Benchmark	No. of Gates	No. of Nets	Near Critical Paths	CSL overhead	Timing Yield without CSL	Timing Yield with CSL
b11	385	322	9	9%	96.5%	99.64%
b12	834	847	16	7.6%	82%	99.65%
b14	4232	4544	65	6.1%	66.2%	99.97%
b15	4585	4716	80	6.9%	48.6%	99.99%
b20	8900	9538	110	4.9%	62.1%	99.95%
b22	12128	13093	118	3.8%	37.0%	99.92%
b17	15524	15911	150	3.8%	56.2%	99.97%
b18	42435	44554	152	1.5%	61.2%	99.99%
Average Percent				5.4%	63.7%	99.9%
Legend: CSL- Clock Stretching Logic Legend: CSL overhead: Percentage of CSL logic area compared to total circuit area Legend: Near Critical Paths: Paths that can violate timing yield with variations						

a simple analysis on selected benchmark circuits to see the impact of clock stretch range on timing yield (Figure 6.7). A smaller value for clock stretch range, for example 5% is shown to impact the timing yield significantly. Hence, with the dual objective of near perfect timing yield and zero short path failures, we have selected the clock stretch range to be 10% of the clock period. In addition to the timing yield improvement results, we have also specified the benchmark characteristics (number of gates and interconnects), the number of near critical paths and the area overhead due to CSL logic in Table 6.2. The proposed CSL methodology also incurs an average area overhead of 5%. The area overhead can be further reduced, if we resort to isolating critical paths similar to the previous works on dynamic clock stretching [34, 76].

6.5 Conclusion

In this chapter, we have proposed a dynamic clock stretching technique to improve the timing yield of circuits in the presence of uncertainty due to process variations. Statistical optimization based techniques due to their over design property, consume extra resources (performance and/or power) even in the absence of variations. The proposed methodology on the other hand, adds timing slack/margin (clock stretching) only in the presence of variations. Further, even in the presence of variations, the proposed methodology activates clock stretching logic only on input patterns that enable the worst critical paths. The dynamic delay detection circuitry, improves yield by controlling the instance of

data capture in critical path memory flops. Experimental results based on Monte-Carlo simulation indicate sizeable improvement in average timing yield with a negligible area overhead.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

The device and interconnect scaling in CMOS circuits with the objective to follow Moore's curve have brought out numerous issues for design, test and manufacturing engineers. The level of miniaturization and integration of billion transistors on a single chip [32], gives a clear picture of the nanometer circuit complexity. The increasing integration levels is introducing new issues, which is making multi-metric circuit optimization more complex. The downward scaling of technology is also gradually reaching the limits of ballistic transportation. Hence, it is crucial to develop circuit optimzation techniques in the nanometer era, that can achieve high performance, low power dissipation and high reliablity. The optimization objectives are highly correlated and conflicting in nature. Further, with increasing levels of variations in process parameters, performance is greatly affected leading to yield loss. It is a challenging task to address all these issues in a single framework. The focus of this dissertation is to address these concerns, by proposing new techniques for modeling and optimiziation of nanometer VLSI circuits considering process variations.

Several researchers have proposed timing analysis based iterative techniques to solve the variation aware circuit optimization problem. The methods, however have a prohibitive runtime. Secondly, the probability distribution based techniques in this area needs detailed statistical infromation on the randomness of the variations. The device level manufacturing tests on fabricated circuits suggest that the uncertainty due to variations in process parameters does not follow any specific distribution. Hence, in this research we propose the use of a fuzzy mathematical programming based optimization technique for variation aware circuit optimization. The uncertainty in delay due to process variations are modeled as interval valued fuzzy numbers with linear membership functions. In specific, we have proposed solutions for the following problems:

I: A layout level gate sizing framework for simultaneous optimization of delay, power, noise and timing yield using a fuzzy linear programming approach

II: A post layout timing based placement technique to optimize delay and timing yield using the concepts of fuzzy nonlinear programming

III: A stochastic chance constrained programming based technique for timing based placement to optimize delay and timing yield at the layout level

IV: A circuit-wise buffer insertion and driver sizing at the layout level using the concepts of fuzzy piece-wise linear programming to optimze power, delay, noise and timing yield

V: A logic level buffer insertion and driver using a look-up table based net length prediction using fuzzy piece-wise linear programming to optimize power, delay and timing yield

VI: A dynamic delay detection and variation compensation using clock stretching to avoid timing failure due to process variations.

The variation aware gate sizing technique is formulated as a fuzzy linear program and the uncertainty in delay due to process variations is modeled using fuzzy numbers. The fuzzy numbers in all the problems use linear membership functions for simplicity. The process variation aware incremental timing based placement problem is modeled as a nonlinear programming problem due to the quadratic dependence of delay on interconnect length. The variation aware timing based placement problem is solved using nonlinear FMP and Stochastic CCP formulations. The stochastic and fuzzy problem provide comparable solutions for the timing based placement problem. In the mathematical programming based variation resistance improvement, we have also proposed a piece-wise linear solution to the buffer insertion and driver sizing (BIDS) problem. The BIDS problem is also solved at the logic level, with look-up table based approximation of net lengths for variation awareness early in the design flow. The logic and layout level solutions are comparable confirming the efficiency of the proposed methodology. Finally, we have proposed a new technique that can dynamically enable variation compensation. A dynamic delay detection circuit is used to identify the uncertainty in delay due to variations. The delay detection circuit controls the instance of data capture in critical path memory flops to avoid a timing failure only in the presence of variations. The proposed methodology improves the timing yield of the circuit with over compenstation only when required. In summary, the various formulation and solution techniques developed in this dissertation achieve significantly better optimization and run times compared to other related methods. The techniques lack any assumption

on the variation parameters and hence can be used to model variations early in the design flow. The proposed methods have been rigorously tested on medium and large sized benchmarks to establish the validity and efficacy of the solution techniques.

Based on the results presented in this dissertation future work could be to integrate the proposed solutions in a single framework and develop new fuzzy mathematical programming based techniques for variation aware multi-metric optimization.

I: The variation aware techniques for gate sizing, buffer insertion and placement can be integrated into a single framework to simultaneous consider the best possible option in each step of the optimization.

II: The multiple threshold voltage assignment and clock skew minimization problems is inherently suited to mathematical programming based formulation. Further, with increasing variations in parameters it is crucial to model the uncertainty in threshold voltage. Hence, the problems can be formulated as a fuzzy mathematical program at the layout level with variations modeled as fuzzy membership functions.

III: In this dissertation, we have modeled the variation parameters as a fuzzy numbers with linear membership functions. The uncertainty in delay due to variations in cirucit optimization problems can also be modeled as fuzzy nonlinear membership functions to compare the efficiency of fuzzy techniques in the presence of multiple variation distribution. The variation resistance coefficient λ will have to be changed to represent the nonlinear membership function.

IV: The dynamic delay detection technique proposed in Chapter 6, in addition to avoiding timing failures due to clock stretching, can also be used for adaptive voltage scaling. The error signal generated from the delay detection latch can be forwarded to the voltage controller to adjust the voltage. If the error signal indicates that multiple critical paths have been affected, and clock stretching occurs too many times, then adaptive voltage scaling can be a better option. The problem needs further investigation.

REFERENCES

[1] H. Tennakoon and C. Sechen. Gate Sizing using Lagrangian Relaxation Combined with a Fast Gradient based Preprocessing Step. In *International Conference on Computer Aided Design*, pages 395–402, 2002.

[2] J.P. Fishburn and A.E. Dunlop. TILOS A Posynomial Programming Approach to Transistor Sizing. In *International Conference on Computer Aided Design*, pages 326–336, 1985.

[3] T. Kuroda, K. Suzuki, S. Mita, T. Fujita, F. Yamane, F. Sano, A. Chiba, Y. Watanabe, K. Matsuda, T. Maeda, T. Sakurai and T. Furuyama . Variable Supply Voltage Scheme for Low-power High-speed CMOS Digital Design. In *IEEE Journal of Solid-State Circuits* , pages 33(3) 454–462, 1993.

[4] A. Agarwal, D. Blaauw, V. Zolotov, S. Sundareswaran, M. Zhao, K. Gala and R. Panda . Statistical Delay Computation Considering Spatial Correlations. In *Asia South Pacific Design Automation Conference*, pages 271–276, 2003.

[5] A. B. Kahng, C. Park, P. Sharma and Q. Wang. Lens Aberration Aware Timing-driven Placement. In *Design Automation and Test in Europe*, pages 890–895, 2006.

[6] A. B. Kahng, S. Mantik and I. L. Markov. Min-max placement for Large Scale Timing Optimization. In *International Symposium on Physical Design*, pages 143–148, 2002.

[7] A. Charnes and W.W. Cooper. Deterministic Equivalents for Optimizing and Satisfying under Chance Constraint. In *Operations Research*, pages 18–39, 1963.

[8] A. Charnes, W.W. Cooper and G.H. Symonds. Cost Horizons and Certainty Equivalents: An Approach to Stochastic Progrmaming of Heating Oil. In *Management Science*, pages 235–263, 1958.

[9] A. Chowdhary, K. Rajagopal, S. Venkatesan, T. Cao, V. Tiourin, Y. Parasuram and B. Haplin. How Accurately can We Model Timing in a Placement Engine? In *Design Automation Conference*, pages 801–806, 2005.

[10] A. Davoodi and A. Srivastava. Variability Driven Gate Sizing for Binning Yield Optimization. In *Design Automation Conference*, pages 294–299, 2005.

[11] A. Devgan and C. Kashyap. Block-based Static Timing Analysis with Uncertainty. In *International Conference on Computer Aided Design*, pages 607–614, 2003.

[12] A. Devgan, and S. Nassif. Power Variability and its Impact on Design. In *International Conference on VLSI Design*, pages 679–682, 2005.

[13] A. K. Murugavel and N. Ranganathan. Gate Sizing and Buffer Insertion using Economic models for Power Optimization. In *International Conference on VLSI Design*, pages 195–200, 2004.

[14] A. Kahng and Y. Pati. Subwavelength Optical Lithography: Challenges and Impacts on Physical Design. In *International Symposium on Physical Design*, pages 112–119, 1999.

[15] A. Srivastava, D. Sylvester and D. Blaauw. Statistical Optimziation of Leakage Power Considering Process Variations using Dual-Vth and Sizing. In *Design Automation Conference*, pages 773–778, 2004.

[16] C. J. Alpert, M. Hrkic, and S.T. Quay. A fast algorithm for identifying good buffer insertion candidate locations. In *International Symposium on Physical design*, pages 47–52, 2004.

[17] M Berkelaar and J. Jess. Gate Sizing in MOS Digital Circuits with Linear Programming. In *European Design Automation Conference*, pages 217–221, 1990.

[18] C. Chen and M. Sarrafzadeh . Simultaneous Voltage Scaling and Gate Sizing for Low-Power Design. In *IEEE Transactions on Circuits and Systems*, pages 400–408, 2002.

[19] C. Hwang and M. Pedram. Timing-Driven Placement Based on Monotone Cell Ordering Constraints. In *Asia South Pacific Design Automation Conference*, 2006.

[20] C. Schmidt and I.E. Grossman. The Exact Overall Time Distribution of Uncertain Task Durations. In *European Journal of Operational Research*, 2000.

[21] C. Visweswariah. Optimization Techniques for High-Performance Digital Circuits. In *International Conference on Computer Aided Design*, pages 198–207, 1997.

[22] C. Visweswariah. Death Taxes and Failing Chips. In *Design Automation Conference*, pages 343–347, 2003.

[23] C.N. Sze, C.J. Alpert, J. Hu and W. Shi. Path based buffer insertion. In *Design Automation Conference*, pages 509–514, 2005.

[24] D. Ernst, N. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner and T. Mudge. A Low Power Pipeline Based on Circuit Level Timing Speculation. In *IEEE/ACM International Symposium on Microarchitecture*, pages 7–18, 2003.

[25] D. Sinha and H. Zhou . Gate Sizing for Crosstalk Reduction under Timing Constraints by Lagrangian Relaxation. In *International Conference on Computer Aided Design*, pages 14–19, 2004.

[26] D. Sinha, N.V. Shenoy and H. Zhou. Statistical Timing Yield Optimization by Gate Sizing. In *IEEE Transactions on VLSI Systems*, pages 1140–1146, 2006.

[27] D. Wang. Meeting Green Computing Challenges. In *Internationaal Symposium on High Density Packaging and Microsystem Integration*, pages 1–4, 2007.

[28] G. N. Rafail and K. Yenilmez. Fuzzy Linear Programming Problems with Fuzzy Membership Functions. In *Mathematical Subject Classification*, pages 375–396, 2000.

[29] H.R. Kheirabadib, M.S. Zamani and M. Saeedi. An efficient analytical approach to path-based buffer insertion. In *IEEE Computer Society Annual Symposium on VLSI*, pages 219–224, 2007.

[30] I. Karkowski. Architectural Synthesis with Possibilistic Programming. In *International Conference on System Sciences*, pages 14–22, 1995.

[31] I. Liu and A. Aziz. Meeting Delay Constraints in DSM by Minimal Repeater Insertion. In *Design, Automation and Test in Europe*, pages 436–440, 2000.

[32] Intel Corporation. Worlds First Two Billion Transistor Microprocessor. In *http://www.intel.com/technology/architecture-silicin/2billion.htm*, 2008.

[33] J. Semiao, J.J. Rodriguez-Andina, F. Vargas, M. Santos, I. Teixeira and P. Teixeira. Improving the Tolerance of Pipeline Based Circuits to Power Supply or Temperature Variations. In *International Symposium on Defect and Fault-Tolerance in VLSI Systems*, pages 303–311, 2007.

[34] J. Semiao, J.J. Rodriguez-Andina, F. Vargas, M. Santos, I. Teixeira and P. Teixeira. Process Tolerant Design Using Thermal and Power-Supply Tolerance in Pipeline Based Circuits. In *IEEE workshop on Design and Diagnostics of Electronic Circuits and Systems* , pages 1–4, 2008.

[35] J. Singh, V. Nookala, Z. Luo and S. Sapatnekar. Robust Gate Sizing by Geometric Programming. In *Design Automation Conference*, pages 315–320, 2005.

[36] J. Xiong and L. He. Fast Buffer Insertion Considering Process Variations. In *International Symposium on Physical Design*, 2006.

[37] J. Xiong, V. Zolotov and L. He. Robust Extraction of Spatial Correlations. In *IEEE Transactions on Computer Aided Design*, pages 619–631, 26(4) , 2007.

[38] J.B. Martins, F. Moraes and R. Reis. Interconnection Length Estimation at Logic-level. In *Integrated Circuits and Systems Design*, pages 98–102, 2001.

[39] J.F. Tang, D. W. Wang, Y.K. Fung, and K.L. Yung. Understanding of Fuzzy Optimization: Theories and Methods. In *Journal of System Science and Complexity*, pages 117–136, 2004.

[40] J.J. Buckley. Stochastic versus Possibilistic Programming. In *International Journal on Fuzzy Sets and Systems*, pages 173–177, 34(2), 1990.

[41] K. Bhattacharya and N. Ranganathan. Reliability-centric Gate Sizing with Simultaneous Optimization of Soft Error Rate, Delay and Power. In *International Symposium on Low Power Electronic Design*, pages 99–104, 2008.

[42] K. Chopra, S. Shah, A. Srivastava, D. Blaauw and D. Sylvester. Parametric Yield Maximization using Gate Sizing Based on Efficient Statistical Power and Delay Gradient Computation. In *International Conference on Computer Aided Design*, pages 1023–1028, 2005.

[43] K. J. George and B. Yuan. Fuzzy Sets and Fuzzy Logic, by Prentice Hall, 1995.

[44] L. Cheng, J. Xiong and L. He. Non-Linear Statistical Static Timing Analysis for Non-Gaussian Variations Sources. In *Design Automation Conference*, pages 250–255, 2007.

[45] L. Deng and M. D. F. Wong. Buffer Insertion Under Process Variations for Delay Minimization. In *IEEE/ACM International Conference on Computer-aided Design*, pages 317–321, 2005.

[46] L. Wei, K. Roy and C. Koh. Mixed-vth CMOS Circuit Design Methodology for Low Power Applications. In *Custom Integrated Circuits Conference*, pages 413–416, 2000.

[47] L.P.P.P van Ginneken. Buffer placement in distributed rc-tree networks for minimal elmore delay. In *International Symposium on Circuits and Systems*, pages 190–197, 1990.

[48] M. Elgebaly and M. Sachdev. Variation-Aware Adaptive Voltage Scaling System. In *IEEE Transactions on Very Large Scale Integration (TVLSI)* , pages 560–571, 15(5), 2007.

[49] M. Hashimoto and H. Onodera. A Performance Optimization Method by Gate Sizing using Statistical Static Timing Analysis. In *International Symposium on Physical Design*, pages 111–116, 2000.

[50] M. Hrkic, J. Lillis and G. Beraudo. An Approach to Placement Coupled Logic Replication. In *Design Automation Conference*, pages 711–716, 2004.

[51] M. Inuiguchi and J. Ramik. Possibility Linear Programming: A Brief Review of Fuzzy Mathematical Programming and Comparison with Stochastic Programming in Portfolio Selection Problem. In *Fuzzy Sets and Systems*, pages 3–28, 2000.

[52] M. Lundstrom. Elementary Scattering Theory of the Si MOSFET. In *IEEE Electron Device Letters*, pages 361–363, 18(7), 1997.

[53] M. Mani, A. Devgan, and M. Orshansky. An Efficient Algorithm for Statistical Minimization of Total Power under Timing Yield Constraints. In *Design Automation Conference*, pages 309–314, 2005.

[54] M. Mani and M. Orshansky. A New Statistical Optimization Algorithm for Gate Sizing. In *International Conference on Computer Design*, pages 272–277, 2004.

[55] M. R. Guthas, N. Venkateswaran, C. Visweswariah and V. Zolotov. Gate Sizing using Incremental Parameterized Statistical Timing Analysis. In *International Conference on Computer Aided Design*, pages 1029–1036, 2005.

[56] M. Sakawa. Genetic Algorithms and Fuzzy Multiobjective optimization, by Kluwer Academic Publishers, 2002.

[57] M.E. Levitt. Design for Manufacturing? Design for Yield! In *International Symposium on Quality Electronic Design*, pages 19–19, 2004.

[58] N. Hanchate and N. Ranganathan. Statistical Gate Sizing for Yield Enhancement at Post Layout Level. In *International Symposium on Very Large Scale Integration*, pages 225–232, 2007.

[59] N. Ranganathan, U. Gupta and V. Mahalingam. Simultaneous Optimization of Total Power, Crosstalk Noise and Delay Under Uncertainty. In *Great Lakes Symposium on VLSI*, pages 171–176, 2008.

[60] O. Coudert . Gate Sizing for Constrained Delay/Power/Area Optimization. In *IEEE Transactions on VLSI Systems*, pages 465–472, 1997.

[61] P. Pant, V. De and A. Chatterjee. Device-circuit Optimization for Minimal Energy and Power Consumption in CMOS Random Logic Networks. In *IEEE Transactions on VLSI*, pages 390–394, 9(2), 2001.

[62] P. Saxena, N. Menezes, P. Cocchini, D.A. Kirkpatrick. Repeater scaling and its impact on cad. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 451–463, 23(4), 2004.

[63] R. B. Hitchcock. Timing Verification and the Timing Analysis Program. In *Design Automation Conference*, pages 594–604, 1982.

[64] R. E. Bellman and L. A. Zadeh. Decision Making in Fuzzy Environment. In *Journal of Management Science*, pages 141–164, 17(4), 1970.

[65] R. Fourer and D. M. Gay and B. W. Kernighan. AMPL: A Modeling Language for Mathematical Programming, by Duxbury Press, 2002.

[66] R. H. Byrd, J. Nocedal, and R. A. Waltz. KNITRO: An Integrated Package for Nonlinear Optimization. In *Large-Scale Nonlinear Optimization (http://www-neos.mcs.anl.gov/neos/solvers/nco:KNITRO/AMPL.html)*, pages 35–59, 2006.

[67] R.B. Lin and M.C. Wu. A New Statistical Approach to Timing Analysis of VLSI Circuits. In *International Conference on VLSI Design*, pages 507–513, 1998.

[68] R.H.J.M Otten and R.K. Brayton. Planning for Performance. In *Design Automation Conference*, pages 122–127, 1998.

[69] S. Bhardwaj, Y. Cao and S.B.K. Vrudhula. Statistical Leakage Minimization Through Joint Selection of Gate Sizes Gate Lengths and Threshold Voltage. In *Asia South Pacific Design Automation Conference*, pages 953–958, 2006.

[70] S. Borkar. Design Challenges of Technology Scaling. In *IEEE MICRO*, pages 23–29, 1999.

[71] S. Borkar, T. Karnik and V. De. Design and Reliability Challenges in Nanometer Technologies. In *Design Automation Conference*, pages 75–75, 2004.

[72] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi and V. De. Parameter Variations and Impact on Circuit and Microarchitecture. In *Design Automation Conference*, pages 338–342, 2003.

[73] S. Boyd and L. Vandenberghe. Convex Optimization, by Cambridge University Press, 2008.

[74] S. Devadas, H. F. Jyu, K. Keutzer and S. Malik. Statistical Timing Analysis of Combinational Circuits. In *International Conference on Computer Design*, pages 38–43, 1992.

[75] S. Dhar, D. Maksirnovi and B. Kranzen. Closed-Loop Adaptive Voltage Scaling Controller for Standard-Cell ASICs. In *IEEE symposium on Low Power Electronic Design*, pages 103–107, 2002.

[76] S. Ghosh, S. Bhunia and K. Roy. CRISTA: A New Paradigm for Low-Power, Variation-Tolerant, and Adaptive Circuit Synthesis Using Critical Path Isolation. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 1947–1956, 26(11), 2007.

[77] S. Nassif. Delay Variability: Sources, Impacts and Trends. In *International Solid States Circuits Conference*, pages 368–369, 2000.

[78] S. Neiroukh and X. Song. Improving the Process-Variation Tolerance of Digital Circuits using Gate Sizing and Statistical Techniques. In *Design Automation and Test in Europe*, pages 959–964, 2006.

[79] S. R. Nassif. The Impact of Variability on Power. In *International Symposium on Low Power Electronic Design*, pages 350–350, 2004.

[80] S. Sapatnekar, V. Rao, P. Vaidya and S. Kang. An Exact Solution to the Transistor Sizing Problem for CMOS Circuits Using Convex Optimization. In *IEEE Transactions on Computer-Aided Design*, pages 1621–1634, 12(11), 1993.

[81] S. Srinivasan and V. Narayanan. Variation Aware Placement for FPGAs. In *International Symposium on VLSI*, pages 422–425, 2006.

[82] T. Burd, T. Pering, A. Stratakos and R. Brodersen. A Dynamic Voltage Scaled Microprocessor System. In *IEEE Solid-State Circuits Conference (ISSCC)* , pages 294–295, 2000.

[83] T. Luo, D. Newmark and D. Z. Pan. A New LP based Incremental Timing Driven Placement for High Performance Designs. In *Design Automation Conference*, pages 1115–1120, 2006.

[84] V. Mahalingam and N. Ranganathan. Variation Aware Timing Based Placement Using Fuzzy Programming. In *International Symposium on Quality Electronic Design*, pages 327–332, 2007.

[85] V. Mahalingam and N. Ranganathan. A Fuzzy Optimization Approach for Process Variation Aware Buffer Insertion and Driver Sizing. In *IEEE International Symposium on VLSI*, pages 329–334, 2008.

[86] V. Mahalingam and N. Ranganathan. Timing Based Placement Considering Uncertainty due to Process Variations. In *Accepted for Publication (Sep 2008) IEEE Transactions on VLSI Systems*, page , 2008.

[87] V. Mahalingam, N. Ranganathan and J. E. Harlow. A Novel Approach for Variation Aware Power Minimization during Gate sizing. In *International Symposium on Low Power Electronic Design*, pages 174–179, 2006.

[88] V. Mahalingam, N. Ranganathan and J. E. Harlow. Fuzzy Optimization Approach for Gate Sizing in the Presence of Process Variations. In *IEEE Transactions on VLSI Systems*, pages 975–984, 16(8), 2008.

[89] V. Mehrotra, S. Sam, D. Boning, A. Chandrakasan, R. Val1ishayee, and S. Nassif. A Methodology for Modeling the Effects of Systematic Within-Die Interconnect and Device Variation on Circuit Performance. In *Design Automation Conference*, pages 172–175, 2000.

[90] W. Chen, C.T. Hseih and M. Pedram . Simultaneous Gate Sizing and Placement. In *IEEE Transactions on Computer-Aided Design*, pages 206–214, 2000.

[91] W. Chow and K. Bazargan. Incremental Placement for Timing Optimization. In *International Conference on Computer Aided Design*, pages 463–466, 2003.

[92] W. Maly. Quality of Design from an IC Manufacturing Perspective. In *International Symposium on Quality Electronic Design*, pages 235–236, 2001.

[93] W. Shi and Z. Li. An o(nlogn) time algorithm for optimal buffer insertion. In *Design Automation Conference*, pages 580–585, 2005.

[94] W.E Donath. Placement and Average Interconnection lengths of Computer Logic. In *IEEE Transactions on Circuits and Systems*, pages 272–277, 26(4), 1979.

[95] X. Bai, C. Visweswariah, S. N. Philip and H. J. David. Uncertainty-Aware Circuit Optimization. In *Design Automation Conference*, pages 58–63, 2002.

[96] Z. Jiang and W. Shi. Circuit-wise buffer insertion and gate sizing with scalability. In *Design Automation Conference*, pages 708–713, 2008.

[97] Z. Ren, D. Z. Pan and D. S. Kung. Sensitivity Guided Net Weighting for Placement Driven Synthesis. In *International Symposium on Physical Design*, pages 124–131, 2004.

ABOUT THE AUTHOR

Mahalingam Venkataraman (V. Mahalingam)(S'03) received the B.E. degree in Computer Science from Sri Venkateswara College of Engineering (SVCE), University of Madras, India, in 2003 and the Master's degree in Computer Engineering from the University of South Florida, Tampa, in 2005, where he is currently working toward the Ph.D. degree in Computer Science and Engineering. In 2002 and 2003, he worked as a part time research assistant at Waran Research Foundation (WARFT), India. In 2008, he worked as a student intern in the ASIC Design for Test (DFT) group at Texas Instruments, Dallas. His research interests include design automation, circuit design, VLSI testing and Design for Manufacturability. Mahalingam is a student member of the IEEE and IEEE Computer Society and has received the IEEE Richard E. Merwin scholarship award in 2006 for excellence in USF-IEEE-CS leadership. He has co-authored over 15 papers in refereed journals and conferences and has actively been involved as a reviewer in TVLSI, TCAD, TODAES, DAC, VLSID, ISLPED, ISQED and ISVLSI.