2006

A game theoretic framework for interconnect optimization in deep submicron and nanometer design

Narender Hanchate
University of South Florida

A Game Theoretic Framework for Interconnect Optimization in Deep Submicron and Nanometer Design

by

Narender Hanchate

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: N. Ranganathan, Ph.D.
Justin E. Harlow III, M.S.
Soontae Kim, Ph.D.
Michael Kovac, Ph.D.
Natasha Jonoska, Ph.D.

Date of Approval:
March 24, 2006

Keywords: Game theory, Crosstalk noise, Interconnect delay, Process variations, Delay uncertainty, Transmission line models, Wire sizing, Gate sizing

DEDICATION

To my wonderful parents,
Ganesh Rao and Annapurna Bai Hanchate,
with all my love and respect

ACKNOWLEDGEMENTS

My sincere thanks to my advisor, Professor N. Ranganathan for giving me an opportunity to work in this very interesting area. I am most grateful to him for his continuous encouragement and valuable suggestions from his vast experience. I consider myself extremely fortunate for having worked with such a prominent leader in the field of VLSI CAD and Algorithms. His teaching, leadership and guidance have been instrumental in my academic and professional development. I would also like to thank Professors Justin E. Harlow III, Dr. Soontae Kim, Dr. Michael Kovac, and Dr. Natasha Jonoska for serving on my committee.

A special thanks and acknowledgement to Mr. Robert Tufts, Assistant Director of NNRC and a wonderful person I have known, for his personal support and valuable suggestions during my course as graduate assistant in NNRC. I would like to acknowledge the efforts of all the contributors who have developed ASIC cores and made them available online for download through Opencores. Acknowledgments to Cadence design systems, Synopsys Inc., developers of Crete program for their tools and 180nm standard cell library, developers of “GALib - a C++ genetic algorithm library” and to Andrew Conn, Nick Gould and Philippe Toint for providing “LANCELOT: a package for large scale non-linear optimization” for open use.

I am very grateful for the invaluable support and motivation that I recieved from my family without which this work would not have been possible. They have taken all the pains on themselves and enabled me to work without distractions. I would also like to thank my friends Viswa, Bheem, Ninnu, Satish, Sandeep, Sridhar and Mouli for their support during my stay at USF.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

A GAME THEORETIC FRAMEWORK FOR INTERCONNECT OPTIMIZATION IN DEEP SUBMICRON AND NANOMETER DESIGN

Narender Hanchate

ABSTRACT

The continuous scaling of interconnect wires in deep submicron (DSM) circuits result in increased interconnect delay, power and crosstalk noise. In this dissertation, we address the problem of multi-metric optimization at post layout level in the design of deep submicron designs and develop a game theoretic framework for its solution. Traditional approaches in the literature can only perform single metric optimization and cannot handle multiple metrics. However, in interconnect optimization, the simultaneous optimization of multiple parameters such as delay, crosstalk noise and power is necessary and critical. Thus, the work described in this dissertation research addressing multi-metric optimization is an important contribution.

Specifically, we address the problems of simultaneous optimization of interconnect delay and crosstalk noise during (i) wire sizing (ii) gate sizing (iii) integrated gate and wire sizing, and (iv) gate sizing considering process variations. Game theory provides a natural framework for handling conflicting situations and allows optimization of multiple parameters. This property is exploited in modeling the simultaneous optimization of various design parameters such as interconnect delay, crosstalk noise and power, which are conflicting in nature. The problem of multi-metric optimization is formulated as a normal form game model and solved using Nash equilibrium theory. In wire sizing formulations, the net segments within a channel are modeled as the players and the range of possible wire sizes forms the set of strategies. The payoff function is modeled as (i) the geometric mean of interconnect delay and crosstalk noise and (ii) the weighted-sum of interconnect delay, power and crosstalk noise, in order to study the impact of different cost functions with two and three metrics respectively. In gate sizing formulations, the range of possible gate sizes is modeled as the set of strategies and the payoff function is modeled as the geometric mean of interconnect delay and crosstalk noise. The gates are modeled as the players while performing gate sizing, whereas, the interconnect delay and crosstalk noise are modeled as players for integrated wire and gate sizing framework as well as for statistical gate sizing under the impact of process variations.

The various algorithms proposed in this dissertation (i) perform multi-metric optimization (ii) achieve significantly better optimization and run times than other methods such as simulated annealing, genetic search, and Lagrangian relaxation (iii) have linear time and space complexities, and hence can be applied to very large SOC designs, and (iv) do not require rerouting or incur any area overhead. The computational complexity analysis of the proposed algorithms as well as their software implementations are described, and experimental results are provided that establish the efficacy of the proposed algorithms.

CHAPTER 1

INTRODUCTION

In deep submicron (DSM) circuits, the interconnects have become a dominant factor in determining the overall circuit performance, reliability, and cost. The increase in the integration density and the chip area results in the increase of total wire length per unit area and decrease in interconnect pitch. According to the International Technology Roadmap for Semiconductors (ITRS) released in 2004, the feature size will continue to scale down at the rate of 0.7× per generation to reach 22nm by 2008 [1]. This reduction rate enforces an increase in impurity concentration and the scaling down of supply and threshold voltages to maintain the electric fields in the device. As the supply voltage is scaled, the interconnect dimensions must also be reduced to take advantage of the feature size scaling [2]. Hence, the combined effect of chip size growth and scaling results in rapid increase of capacitance and resistance of interconnect wires. This increases the propagation delay through interconnects by a factor of $S^2 S_C^2$, where S is the scaling factor and S_C is the chip size increase factor which accounts for the increase in chip size from one generation of ICs to the next generation [3].

The interconnect effects like the rising RC delay of on-chip wiring, noise considerations such as crosstalk and delay unpredictability, uncertainty due to process variations, reliability concerns due to rising current densities and oxide electric fields, and increasing power dissipation are becoming increasingly prominent in deep submicron and nanometer designs [4, 5]. The logic cell delays have reached picosecond range and continue to reduce due to the scaling of the minimum feature size. However, the interconnect delays are increasing and are capable of consuming the majority of the clock cycle time in DSM designs [6]. The main reason for increased wire delays in DSM is the increase in resistance, which is inversely proportional to its cross-sectional area. Interconnect capacitances are also increasing due to higher wiring densities needed to route DSM chips. The noise in digital circuits is defined as any deviation of a signal from its stable value in those intervals where it should have been stable otherwise. In deep submicron designs, the main sources for noise are interconnect cross-capacitance due to coupling, power supply fluctuations, mutual inductance and thermal noise due to self-heating caused by current flow [7].

The interconnect capacitance has three components: area or ground capacitance, fringing field capacitance and coupling capacitance. The high aspect ratio of wires result in more wire to wire capacitance among the neighboring wires in the same layer than the area capacitance between upper/lower wiring layers [7]. In addition, wire space scaling due to increased wiring densities also increases coupling capacitance. The interconnect cross-capacitance noise is due to the charge injected in quiet/silent nets because of switching in neighboring nets through the coupling capacitance between them. The charge injected increases prominently in the deep submicron regime due to the increased coupling capacitance between adjacent nets causing reliability issues [8]. The noise due to coupling capacitance is the dominant component among the noise sources and is a major concern in deep submicron design [9]. Power supply noise is the spurious signal appearing at the receiver due to the difference between the local reference voltage levels at the driver and the receiver. There are two components of power supply noise: low frequency and high frequency. The low frequency component is known as IR drop and is due to the variation in the DC power supply and ground levels. The simultaneous switching of various sub-circuit modules produce the high frequency component of power supply noise generally called as delta noise. Mutual inductance noise is caused by the change in the magnetic field due to the transient current flow through the loop formed by the signal wire and the current return path. Thermal noise is due to the self heating caused by the current flow in the interconnects, limiting the maximum allowed average current density.

The noise causes delay and functional failures due to Miller effects and signal deviations, and increases the power consumption due to glitches. A static wire called the victim is perturbed by the switching activity on the neighboring wires called aggressors. When the aggressors switch in same direction simultaneously, an undesirable voltage spike is coupled on to the victim due to capacitive coupling causing false switching or voltage overshoot affecting signal integrity [10]. Hence, it is important to design noise immune DSM circuits considering the continuous trend of scaling of interconnect dimensions. Modeling the noise of a circuit will need complete information of nets (its neighboring nets, the length of overlap, spacing between the nets, etc) to analyze the coupling effects, and hence, is typically performed after the final routing of the design [11]. The standard methods practiced to reduce the DSM effects due to interconnects are driver sizing, buffer sizing, buffer insertion, wire sizing, wire spacing, net ordering and wire shielding.

The aggressive scaling of VLSI technology has given rise to increased impact of process variations on the performance, reliability and power of the fabricated circuits. Limitations due to the manufacturing processes and environmental noise degrade the quality of signals and affect the propagation delay of the circuit [4]. These effects force the propagation delay to deviate from its typical or nominal value, resulting in delay unpredictability. This deviation of the propagation delay due to delay unpredictability is defined as delay uncertainty. The examples of factors which cause delay uncertainty are non-uniformity of gate oxide thickness; imperfections in polysilicon etching, photolithography, planarization and metal etching processes; and environmental noise due to changes in ambient temperature and external radiation [12]. Uncertainty in propagation delay of signals can cause violations in set-up and hold timing constraints, resulting in timing failure of the design. To eradicate these timing violations, the designer has to relax the timing constraints or has to reduce the delay uncertainty. Relaxing timing constraints increases the clock period and hence degrades the circuit performance. Thus, the designer has to reduce the delay uncertainty in order to meet the tight timing constraints of the design without compromising the performance of the design.

The device and interconnect scaling trends make the physical realization of devices and interconnects unpredictable during front-end design, hence changing the paradigm of the design problem from deterministic to probabilistic domain [5]. In addition, the process tolerances do not scale proportionally, thereby causing the relative impact of the process variations to increase dramatically with every new technology generation. The performance of an integrated circuit is impacted by two distinct sources of variation [13]: (i) environmental factors which include variations in power supply voltage and temperature, (ii) physical factors which include variations in the electrical and physical parameters characterizing the behavior of active and passive devices, caused by processing and mask imperfections. The physical factors can be further classified as: (i) die-to-die physical variations which are largely independent of the design implementation and are usually modeled using worst-case corners, (ii) within-die physical variations such as variations in gate dimension influenced by layout design implementations, and for which worst-case corner modeling is insufficient [5, 14]. The worst-case corner modeling maximizes a single device parameter (e.g. delay, noise or power) and does not usually take the spatial correlations into account, thereby resulting in too pessimistic an approach. As a result, some of the valid designs may be rejected or have to be adjusted to meet artificial and inaccurate worst case limits. This can lead to unnecessarily large chip area and power consumption as well as increasing design efforts and costs.

The physical parameters such as the width, the thickness of the interconnects and the effective length of the MOS devices vary significantly between the intra-die and inter-die components. These physical variations lead to substantial variations in the electrical parameters such as conductance, capacitance, inductance, threshold voltages, and leakage currents of the CMOS devices and interconnects. The intra-die variations exhibit spatial correlations, where devices that are close to each other have a higher probability of having similar device properties than those which are placed far apart. Also, when coupled with process variations, noise effects can produce worst-case design corner combinations for the design that represents extreme operating conditions, causing prime reliability concerns. Hence, it is essential for the design tools to account for these uncertainties and design robust circuits that are insensitive to the process variations.

1.1 Motivation: Multi-metric Optimization

To summarize the above discussion, the continuous device and interconnect scaling trends in deep submicron and nanometer designs have resulted in prominent effects like increased interconnect delay due to rising parasitic resistance and capacitance of on-chip wiring, noise and reliability concerns due to crosstalk and coupling capacitance, delay uncertainty due to process variations, and increased interconnect power dissipation. In addition, the physical realization of devices and interconnects are unpredictable due to the random nature of process variations. Thus, the *major challenge* is in achieving reliable, low-power, and high-performance system implementations from the micro-architecture level down to the layout level, considering unpredictable behavior due to process variations. In order to realize such a system implementation, the traditional method of design optimization for numerous years - single metric optimization with other design parameters as constraints, is no longer effective or sufficient. On the contrary, in DSM circuits, it is significantly important to simultaneously optimize various design parameters (interconnect delay, crosstalk noise, delay uncertainty, interconnect power). Hence, there is a need for new methods and algorithms capable of performing multi-metric optimization.

Multi-metric optimization in conflicting environments is a difficult problem since the normal definition of an optimal value no longer applies or valid. For example, an optimal gate size for one metric may not be optimal for another metric. The optimal policy at any given instance depends on the polices for other metrics, keeping the best interest of the entire system in view. While most optimization methods such as ILP, simulated annealing, and force directed methods lend themselves well to single metric optimization, these methods are inadequate for multi-metric optimization. Hence, in this dissertation, we investigate the application of game theory, a multi-agent optimization framework, to the problems of VLSI CAD. We have used game theoretic models to solve the problems of post layout wire and gate sizing for multi-metric optimization without considering process variations. The consideration of process variations during design optimization requires probabilistic analysis due to the uncertainty element introduced by process variations. In order to capture the nondeterministic behavior of system parameters due to uncertainty, we have used stochastic game modeling for solving post layout gate sizing problem considering process variations.

The fundamental basis and structure of game theory and stochastic games allow the formulation of optimization problems in which multiple inter-related cost metrics compete against one another for their simultaneous optimization. There exist several approaches such as the Nash equilibrium and ϵ-equilibrium for achieving equilibrium state solutions in which each metric is optimized with respect to the optimality of others. Further, the stochastic game models inherently capture the nondeterministic behavior of the system parameters due to process variations. These factors make game theory a powerful tool to model optimization problems in VLSI design automation. Game theory supports the following four features: rationality, coalition formation, competition and equilibrium. Game theoretic reasoning takes into account the attempts made by the multiple agents towards the optimization of their objectives for every decision. Each agent or a player's decision is based on the decision of every other player in the game and hence it is possible for each player to optimize his gain with respect to the others' gains in the game. Thus, game theoretic models try to achieve global gain among the set of given players. In terms of game theory, a solution is said to reach its global value for the given conditions when the equilibrium condition is met.

The solutions to game theoretic models exhibit the property of social equilibrium [15], which enforces that the optimization of individual decisions have to take into account the optimization of other players' decisions. In other words, while making decisions towards optimizing one metric may not be able to optimize other metrics. Hence, game theory is a natural framework which inherently considers social equilibrium with respect to the individual decisions as well as the global objective to ensure the fairness objective. Further, if the payoff function is convex, i.e., the parameters being optimized correspond to conflicting objectives (such as delay and crosstalk noise in interconnects), game theoretic optimization, in fact, performs significantly better than methods such as simulated annealing, genetic search and Lagrangian relaxation for problems with conflicting objectives [16, 17, 18].

1.2 Contributions of this Dissertation

This dissertation addresses the problem of multi-metric optimization in the design of deep submicron CMOS VLSI circuits. The following are the major contributions of this dissertation.

(i) Wire sizing: We developed a multi-metric optimization framework for performing wire sizing at post layout level. The framework is based on game theory which can simultaneously optimize (a) interconnect delay and crosstalk noise (b) Interconnect delay, power and crosstalk noise. We have shown that wire sizing is a powerful and effective technique in making use of the unused routing resources while optimizing design parameters at post-route stage. The work reported in the literature on wire sizing perform only delay optimization, and do not consider routing congestion. They result in unconstrained wire sizes which cannot be applied directly for sizing the nets of a routed design. Hence, we develop new algorithms for wire sizing at post layout level which neither need rerouting of nets nor area overhead. The developed algorithms have linear complexity in terms of both time and space.

(ii) Fast interconnect models: New transmission line based interconnect models have been developed for an arrangement of three parallel interconnect wire segments. These models are simple, fast and accurate, and hence, can be used in frameworks which need multiple repetitive calculation based on the analytical models.

(iii) Gate sizing: We developed a new post-layout gate sizing algorithm for simultaneous optimization of interconnect delay and crosstalk noise. The gate sizing problem is modeled using game theory and solved using the Nash equilibrium. We have proposed two different approaches in which the games are ordered according to the noise criticality or the delay criticality of the nets. The developed algorithms are linear in terms of time and space and do not require rerouting or area overhead.

(iv) Gate and wire sizing: We developed an integrated gate and wire sizing framework for simultaneous optimization of interconnect delay and crosstalk noise. The problem is modeled using game theory and solved using the Nash equilibrium. The integrated framework performs the scaling of gate and wire sizes more effectively than the sequential approach of wire sizing followed by gate sizing or gate sizing followed by wire sizing. We have proposed two different approaches in which the games are ordered according to the noise criticality or the delay criticality of the nets. The developed algorithms are linear in terms of time and space and do not require rerouting or area overhead.

(v) Statistical Gate sizing considering process variations: We developed a multi-metric optimization framework for minimizing delay uncertainty and crosstalk noise under the impact of process variations. We have used stochastic games to solve the statistical gate sizing problem. The formulation and the developed framework is completely stochastic. The process parameter distributions are modeled using the stochastic function by controlling the state transition and the payoffs to the players. The approach is independent of probability distributions used to model process variations. In other words, It can work for any statistical distributions like Gaussian or Log-Normal, and hence, can be applied to 65nm designs or below. The developed algorithms do not require rerouting or area overhead.

1.3 Outline of this Dissertation

The remainder of this dissertation is organized as follows: Chapter 2 describes the prior work based on wire and gate sizing with and without the considerations of process variations, some of the relevant concepts of game theory and stochastic games along with their modeling, and the interconnect models used in most part of this dissertation. Chapter

3 defines the problem of post layout wire sizing and provides two different game theoretic formulations for (i) simultaneous optimization of interconnect delay and crosstalk noise and (ii) simultaneous optimization of interconnect delay, power and crosstalk noise. In Chapter 4, we have derived fast interconnect models based on transmission lines and verified them in the context of wire sizing problem. These models are found to be faster than those given in Chapter 2.5 with same level of accuracy. Chapter 5 defines the problem of post layout gate sizing and provides a game theoretic solution with two different strategies in which games are ordered according to (i) the noise criticality and (ii) the delay criticality of the nets. In Chapter 6, the problem of integrated gate and wire sizing for multi-metric optimization is addressed. The game theoretic modeling developed in this Chapter is a two-player game model which is completely different when compared to the respective frameworks for gate and wire sizing problems. Chapter 7 defines the problem of statistical gate sizing considering process variations and develops a stochastic game based Nash equilibrium solution for minimization of delay uncertainty and crosstalk noise. The concluding remarks and future work related to the dissertation are given in Chapter 8.

CHAPTER 2

BACKGROUND AND RELATED WORK

The various issues related to the design of interconnects are discussed in Chapter 1. There has been a significant amount of research in estimating and optimizing the effects due to interconnects. In this chapter, we highlight the research that is focused on optimizing the interconnect effects and in particular, the work which is related to the problems addressed in this dissertation.

Some of the standard procedures practiced to reduce the effects of interconnects in DSM circuits are (i) driver/gate sizing (ii) wire sizing (iii) buffer insertion (iv) net reordering (v) wire spacing (vi) wire shielding. Driver sizing is the process of appropriately sizing the driver gates so as to reduce the interconnect effects. If the driver gates are sized-up, the driving strength of the gate increases and hence, the amount of current driven through the interconnect wire connected at the output of the gate increases. This decreases the time required to charge the output capacitance at the other end of the interconnect and thereby reduces interconnect delay. On the other hand, the charge coupled to the adjacent interconnects through the coupling capacitance also increases due to the increased current on the interconnect driven by the sized-up gate. This increases the magnitude of the crosstalk noise induced on the adjacent nets. Thus sizing the driver gates appropriately can strike a balance between the interconnect delay and crosstalk noise.

Buffer insertion is the phenomenon of introducing high strength buffers on long interconnect wires. On long interconnect wires, the signal gets weaker as it travels through its length because of the parasitic resistance and capacitance of the wire. Inserting buffers at intervals throughout the length of long wires help in restoring the weak signals and thereby moving the signal faster. The insertion of buffers should be done judiciously since their introduction results in area overhead and also increases the crosstalk noise induced on the adjacent wires. It is costly to perform buffer insertion at post layout level because the design will need rerouting. Wire sizing is the process of increasing or decreasing the size of the wires in order to reduce the interconnect effects. Increasing the size of a wire will reduce the parasitic resistance and hence, helps in making the signal move faster without loosing its strength. Hence, the effect on the interconnect delay and crosstalk noise due to sizing up a wire is equivalent to sizing up the driver gates.

Wire reordering is another technique to reduce the interconnect effects in DSM circuits. Reordering is the process of shuffling the wires, thereby changing their adjacencies with respect to other wires and making them less susceptible to crosstalk effects. It effectively changes the length of overlap and spacing of a wire with respect to its neighbors. Wire shielding is one of the simplest techniques to minimize crosstalk. It uses shield wires in between the signal wires which are highly susceptible to crosstalk noise. The shield wires are the new wires inserted in the design which are kept at zero potential, in order to act as an effective barrier between the highly coupled victim and aggressor nets. However, the disadvantage of this technique is that it increases the area overhead due to the large number of shield wires which may be required between victims and aggressors in large designs. In the literature, driver or gate sizing and buffer insertion are the preferred techniques for minimizing interconnect effects, but wire sizing is relatively less investigated. In this dissertation, we focus on the wire sizing and gate sizing techniques with the goal of minimizing the interconnect effects. The problems are first modeled as deterministic formulations without considering process variations, followed by stochastic formulations which consider process variations.

2.1 Wire Sizing

In this section, we present a brief overview of prior work focussed on wire sizing. Figure 2.1. shows the various pioneering work found in the literature for the problem of wire sizing. Early work on wire sizing [19, 20] consider the interconnect wires by dividing each wire into smaller segments and assume each wire segment to be of uniform width. In continuous wire sizing, a variation of uniform wire sizing, each wire is divided into infinitely many segments [21]. These approaches tend to use many discrete or an infinite number of wire widths and hence, make interconnect planning difficult [22]. In [23], the authors have provided closed form solutions to simultaneous buffer insertion/sizing and wire sizing. They have developed mathematical formulae using Elmore delay models for wire sizing by considering the given wire as n net segments. The above works do not explicitly attempt to consider the coupling capacitance between the interconnect wires, which is a major concern in DSM circuits. Also, these works have minimization of delay or area-delay product as their objectives.

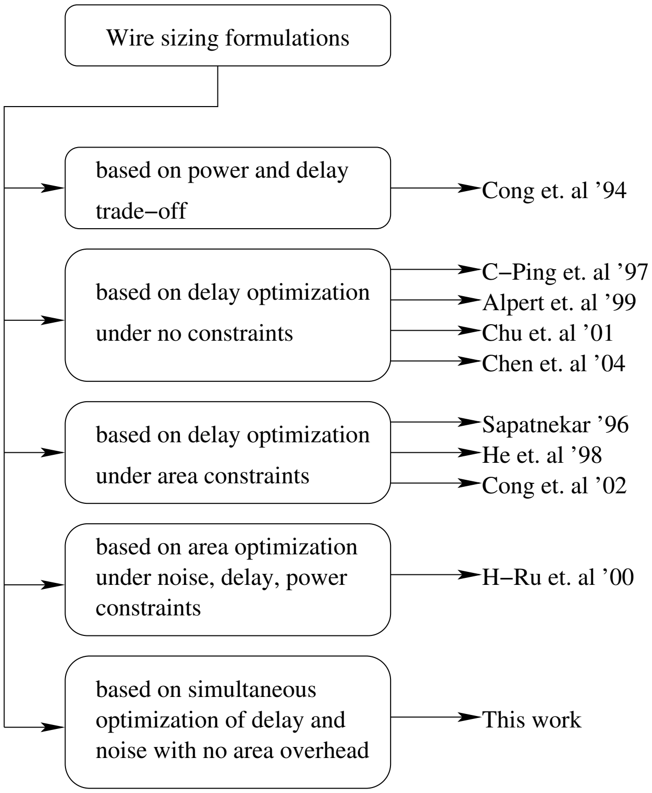

Wire sizing formulations
based on power and delay trade−off
Cong et. al '94
based on delay optimization under no constraints
C−Ping et. al '97
Alpert et. al '99
Chu et. al '01
Chen et. al '04
based on delay optimization under area constraints
Sapatnekar '96
He et. al '98
Cong et. al '02
based on area optimization under noise, delay, power constraints
H−Ru et. al '00
based on simultaneous optimization of delay and noise with no area overhead
This work

Figure 2.1. Taxonomy of Various Pioneering Works on Wire Sizing

The works reported in [24, 22] attempt to consider the coupling capacitance for optimizing the wire geometry. In [24], input probabilities are propagated to obtain the switching conditions of nets. In [22], two simple wire sizing schemes called single-width sizing and two-width sizing are analyzed. These two works result in some inaccuracy as they do not consider the interconnect positions in the final layout, an important factor affecting coupling between the given nets. In addition, the works consider the wires as independent sets to perform wire sizing without taking into account the effects of their surroundings and the final layout conditions such as routing congestion, the set of neighbors, etc. They do not develop design methodologies to generate large feasible designs. In [25], simultaneous wire sizing and wire spacing is performed at post layout with delay minimization as the objective. The above mentioned works use Elmore delay models to model the interconnects and hence lack accuracy [26]. The existing works on wire sizing do not consider the problem of simultaneous optimization of interconnect delay and crosstalk noise with accurate models. In Chapter 3, we develop a complete design framework capable of performing simultaneous optimization of interconnect delay, power and crosstalk noise through wire sizing at post-route level. In Chapter 4, We develop new fast models for interconnects which consider the transmission line behavior of wires to represent the interconnect delay and crosstalk noise of individual nets.

2.2 Gate Sizing

In this section, we present a brief overview of prior works focussed on gate sizing. In one of the pioneering works, Cong et.al. [19] developed a simultaneous gate and wire sizing algorithm for power optimization under delay constraints. In [24], H-Ru et.al. proposed a Lagrangian relaxation based interconnect optimization under the consideration of crosstalk noise. However, they do not consider the exact physical location of the nets in the design. Chu et.al. [23] developed a closed form expression for buffer and wire sizing for delay optimization without considering the crosstalk noise, for use at early stages of design. In [27], Xiao et.al. have used a crosstalk aware static timing analyzer to eliminate the timing violations. In [28], Alpert et.al. have used a delay penalty estimation technique to achieve timing closure. In [29], Albrecht et.al. have developed a linear programming based buffer and wire sizing algorithm for floorplan area minimization under delay constraints. In [30], Hashimoto et.al. have developed a gate sizing algorithm for crosstalk noise optimization under delay constraints which is only capable of down-sizing the aggressor gates. In [10], a greedy gate sizing approach is proposed for minimizing the crosstalk noise by creating a coupling graph with the help of Clarinet noise analyzer. The gates are iteratively sized-up or sized-down to satisfy the noise criterion. In [31], a Lagrangian relaxation based gate sizing approach is proposed for reducing the crosstalk noise under the delay constraints. The algorithm is also iterative and uses a coupling graph extracted based on the coupling capacitances.

In [32], the authors have developed a game theoretic algorithm for gate sizing and buffer insertion at the logic level for power minimization under delay constraints. The developed algorithm is path-based and uses auction theory to implement the delay constraints as a divisible resource. In Chapter 5, we address the problem of simultaneous optimization of interconnect delay and crosstalk noise using gate sizing. We use game theory as an optimization tool to find the optimal gate sizes for the simultaneous reduction of interconnect delay and crosstalk noise. In the game model, we have modeled the gates as the players, its possible gate sizes as the strategy set and the geometric mean of interconnect delay and crosstalk noise as the payoff function. In Chapter 6, we investigated the problem of simultaneous optimization of interconnect delay and crosstalk noise using an integrated approach of gate and wire sizing. We have created a two player normal form game with delay and noise as the players.

2.3 Process Variations

In this section, we present a brief overview of prior works focused on gate sizing with the consideration of process variations. The impact of process variations, their source and their variation trends have been discussed in the pioneering works of Nassif [13, 4, 38] and Borkar et al [5] in great detail. In [39], a statistical design approach is presented to study the impact of interconnect process variations on memory design and performance using Monte Carlo and sensitivity analysis. Monte Carlo analysis shows that the threshold voltage, the effective gate length, the effective gate width and the supply voltage are the key parameters

Table 2.1. Recent Statistical Gate Sizing Works Found in Literature

Recent Works	Objective of Work	Methodology	Assumptions on Process Parameters	Remarks
Jacobs et al 2000 [33]	Minimize delay uncertainty	Non-linear programming	Gaussian distribution	Statistical delay model, uses SSTA, does not consider spatial correlations
Raj et al 2004 [34]	Minimize delay uncertainty	Non-linear programming	Gaussian distribution	defines utility functions, generates statistical delay model, path based, cannot be applied to large circuits
Chopra et al 2005 [35]	Minimize leakage power and delay	Gradient computation	Gaussian distribution	lumps Gaussian distribution of two or more random variables into one - an approximation
Sinha et al 2005 [36]	Minimize delay violations	Heuristic approach SSTA	Gaussian distribution	A set of heuristics with perturbation, builds statistical delay models for gates in library
Singh et al 2005 [37]	tradeoff - area and delay violations	Geometric programming	Any distribution	Posynomial Elmore delay model, uses STA with statistical delay constraints, computationally fast
Our work	Minimize delay and noise violations	Stochastic games	Any distribution	Purely stochastic approach, multi-metric optimization, no area overhead

that influence the interconnect delay, the total average power and the crosstalk noise. Studies on process variations have been mainly focused on variability modeling [14, 34, 40] and statistical static timing analysis (SSTA) [33, 41, 42, 43, 36]. In [33], the statistical gate size optimization problem is solved as a non-linear problem with delays modeled as Gaussian functions, ignoring the spatial correlations due to intra-die variations. In [14], the gate sizing problem is modeled as a deterministic non-linear optimization problem with the help of a penalty function to intentionally improve the timing slacks on non-critical paths of the circuit. In [34], a heuristic approach for statistical gate sizing is proposed for improving the timing yield using the concept of statistically undominated paths. However, this approach is path-based and cannot be applied to large circuits, since as the paths gets larger, the number of gates increases becoming computationally intensive.

The delay uncertainty induced due to process variations on buffer-driven interconnect lines is analyzed in [44] and a buffer sizing methodology is developed to reduce delay uncertainty. In [41], a pruning strategy based on perturbation bounds is developed to solve the statistical gate sizing problem. In [42], the gate sizing problem is solved by identifying the worst negative statistical slack paths with the objective of reducing the performance variance of a technology-mapped circuit. In [43], an incremental and parametric SSTA is proposed to perform gate sizing with pre-targeted yield optimization. In [35], a heuristic approach is provided to compute the gradient of yield with respect to gate sizes and non-linear optimization is performed to maximize the yield. A probabilistic methodology is developed in [45] for buffer insertion problem using a bottom-up recursive approach to calculate the joint probability density function to correlate between arrival times and down-stream capacitance.

In [36], statistical models for the gates in the standard cell library are developed using SSTA by characterizing at different points in the parameter space. As pointed out in [37], the methods based on SSTA need to make (i) the assumptions such as the signal arrival time and the slope have normal distributions, and (ii) the approximations such as the resultant of two or more normal distributions is also a normal distribution, which may be inaccurate. In [37], the uncertainty due to process variations is incorporated in delay constraints using a posynomial delay model and solved for tradeoffs between the area and robustness using geometric programming. This approach is computationally efficient, however does not consider crosstalk violations. Table 2.1. gives a summary of the recent work on statistical gate sizing found in the literature.

2.4 Game Theory and Stochastic Games

In this section, we briefly discuss some of the relevant concepts of game theory and stochastic games. For detailed treatment of these concepts along with various other concepts, please refer to [46, 16] for game theory and to [47, 48] for stochastic games.

2.4.1 Game Theory

Game theory, in a broad sense, can be defined as a collection of mathematical models formulated to analyze the interaction of decision makers in situations of conflict and cooperation. The objective is to find a set of best actions for each decision maker and recognize the corresponding stable outcomes. A game consists of players who choose from a list of alternative courses of action, resulting in outcomes over which the players may have different preferences. Game theory is a guide which implements rational behavior of individual players and predicts their outcomes.

Game theory was formulated as a general theory of rational behavior by von Neumann. The basic building blocks of game theory are based on theories proposed by von Neumann in 1928 [49] and Nash in 1950 [50]. The essential elements of a game are players, actions, payoffs, and information, which are collectively known as the rules of the game. Players of the game are the set of rational decision makers. The goal of each decision maker is to maximize his own utility by a set of actions in the presence of other decision makers. An action or a move by a player i, denoted by a_i, is a choice. The strategy s_i of a player i is a rule to choose an action at each instant of the game. The set of strategies $S_i = \{s_i\}$ available to player i is denoted as his strategy set or strategy space. A strategy combination $\mathbf{s} = (s_1, \ldots, s_N)$ is an ordered set consisting of one strategy for each of the N players in the game.

In an N-player game, the payoff of player i, denoted by $P_i(s_1, \ldots, s_N)$, is the utility obtained after the players had chosen their strategies and the game is played out. It can also be defined as the expected utility received by player i as a function of strategies chosen by each player. The information set of a player i is the knowledge of actions, previously chosen by the players at a given course of the game. The information set of the players changes as the game progresses. An equilibrium is a strategy combination consisting of a best strategy for each of the N players in the game. The equilibrium strategies are the strategies chosen by players to maximize their individual payoffs, among the possible strategy combinations obtained by arbitrarily choosing one strategy per player. The equilibrium outcome is the set of payoff values of the players corresponding to their equilibrium strategies.

Games can be broadly classified into two distinct categories: non-cooperative and cooperative games. In non-cooperative games, players choose their strategies independently and the rules of the game do not allow binding commitments among the players. In other words, non-cooperative games are played with fully rational players. It focuses on the strategies chosen by each player and their respective payoffs. In cooperative games, players form coalition among a subset of players and play their joint actions according to the agreements made during their binding commitments. It focuses on coalition formation and distribution of the benefits gained through cooperation. A general game theoretic model can be classified into three categories based on mathematical formulations - the normal or strategic form, the extensive form and the characteristic function form. The characteristic function form is applicable only to cooperative games.

A non-cooperative game is represented in one of the two general mathematical formulations based on the types of moves employed by the players. The first formulation is the normal form game, in which the players move simultaneously to choose their strategies. In this game, the strategies are same as the actions in ranked coordination. The normal form shows what payoff results from each possible strategy combination, while the outcome matrix shows what outcome results from each action combination. As the players make their moves simultaneously, they do not have a chance to learn each other's private information by observing each other. Thus, in normal form games, the information set of each player about the other players is zero. Therefore, a normal form game is represented by the list of players, their strategy set and the payoff functions. The second formulation is extensive form games which are also called sequential move games. The players of this game move sequentially and choose their strategies according to an order. The order of the play is important and affects the final outcome of the game. The extensive form game is represented by a list of players, their set of actions, information set and payoff functions. The strategies of the players are a series of action moves. The information sets represents the various states each player can take at any given point in the game. The important difference between simultaneous move games and sequential move games is that in sequential move games the second player acquires the information on how the first player moved before he makes his own decision.

A non-cooperative game is called a finite game if the strategy sets $S_1, \ldots, S_N$ are finite. Finite games are given by listing the payoffs for each player in tabular form. A finite game can be generalized to consist of N players who choose from a set of strategies S_i where, $i = 1, ..., N$, and a set of payoff functions P_i where, $i = 1, ..., n : S_1 \times ... \times S_n \rightarrow \Re$, where, $\Re$ is the set of all real numbers and a payoff value is assigned to each pair of strategies chosen by the players. The rationality or the equilibrium point is a set of strategies that maximizes or minimizes the payoff of the player assuming that all other players strategies are held fixed. The game is played until each player's strategy is optimal with respect to the strategies of others. Stackelburg's equilibrium [51] and the Nash equilibrium [52] are some of the techniques which can be used to reach a game's equilibrium. In this work, we focus on non-cooperative finite games.

The Nash equilibrium defined here is in terms of normal form games, which can be easily extended to extensive form games. Let $G = \{S_1, \ldots, S_N; P_1, \ldots, P_N\}$ be a non-cooperative finite game in normal form with N players. The set S_i contains all the strategies and the set P_i contains the corresponding payoff values for a player i. The N-tuple of strategies $\mathbf{s}^* = (s_1^*, \ldots, s_N^*)$, where $s_1^* \in S_1, \ldots, s_N^* \in S_N$, is defined to be Nash equilibrium point of G if

$$P_i(s_1^*, \ldots, s_i^*, \ldots, s_N^*) \geq P_i(s_1^*, \ldots, s_{i-1}^*, s_i, s_{i+1}^*, \ldots, s_N^*)$$

holds $\forall s_i \in S_i$ and $i = 1, \ldots, N$. It can be stated in simple words as once being in the state represented by the strategy choices $\mathbf{s}^*$, the player's payoff does not get better if he unilaterally deviates from the Nash equilibrium strategy. Here, the word unilaterally means that the other players will stick to their equilibrium strategies. The Nash equilibrium point NE, defines the payoff values for all the players in the game. Qualitatively, the Nash equilibrium for an N-player finite game is a N-tuple set of strategies $\mathbf{s}^* = (s_1^*, \ldots, s_N^*)$, given by N inequalities such that, no single player can gain by changing only his strategy.

2.4.2 Stochastic Games

Stochastic games can be defined as the natural and hybrid extension of the Markov Decision Processes (MDPs) and matrix games. Markov decision processes are a single agent and multiple state framework. Matrix games are a multiple agent and single state framework. On the contrary, stochastic games are a multiple agent and multiple state framework, which can be visualized as the merging of MDPs and matrix games. In a multi-agent setting, stochastic games allow the state transition to depend jointly on all the agent actions, and having the immediate rewards at each state determined by a multi-agent general-sum matrix game associated with that state. A N-player nonzero-sum stochastic game is defined as a tuple (N, S, A_i, P, R_i), where N is the number of players, S is the set of states for the game, A_i is the set of actions available to player i, P is the transition probability function $S \times A \times S \rightarrow [1, 0]$, and $R_i : S \times A \rightarrow \Re$ is the payoff or reward function for the i^{th} player. If s is a state at some stage of the game and the players select an $a \in A(s)$, then $p(.|s, a)$ is the probability distribution of the next state of the game. The transition probability p has a density function z with respect to a fixed probability measure μ on S, satisfying the following continuity condition: For any sequence of joint action tuples $a^n \rightarrow a^0$,

$$\int_S |z(s, t, a^n) - z(s, t, a^0)| \mu(dt) \rightarrow 0 \quad as \quad n \rightarrow \infty.$$

In a two-player matrix game scenario, let the matrix pair (M_1, M_2) specify the payoffs for the player 1 (row player) and player 2 (column player), where the matrices M_1 and M_2 are n by n with their indices ranging from 1 to n. If the row player chooses the index i and the column player chooses the index j, then the player 1 receives a payoff of $M_1(i, j)$ and player 2 receives $M_2(i, j)$. The indices i and j are called pure strategies of players 1 and 2 respectively. If α and β are distributions (called mixed strategies) over the row and column indices, the expected payoff to player $k \in \{1, 2\}$ is $M_k(\alpha, \beta) \doteq \mathbf{E}_{i \in \alpha, j \in \beta}[M_k(i, j)]$. The mixed strategy pair (α, β) is said to be a Nash equilibrium for the game (M_1, M_2) if (i) for any mixed strategy α', $M_1(\alpha', \beta) \leq M_1(\alpha, \beta)$, and (ii) for any mixed strategy β', $M_2(\alpha, \beta') \leq M_2(\alpha, \beta)$. A strategy pair (α, β) is defined as the ϵ-Nash, an approximate Nash equilibrium for (M_1, M_2) if (i) for any mixed strategy α', $M_1(\alpha', \beta) \leq M_1(\alpha, \beta) + \epsilon$, and (ii) for any mixed strategy β', $M_2(\alpha, \beta') \leq M_2(\alpha, \beta) + \epsilon$.

A two-player stochastic game G over a state space S consists of a designated start state $s_0 \in S$, a matrix game $(M_1[s], M_2[s])$ for every state $s \in S$, and transition probabilities $P(s'|s, i, j)$ for every $s, s' \in S$, every pure row strategy i, and every pure column strategy j. The stochastic game proceeds as follows: If the game is currently in state s and the two players play mixed strategies α and β, then pure strategies i and j are chosen according to α and β respectively, and the players receive an immediate payoffs of $M_1[s](i, j)$ and $M_2[s](i, j)$. The game then moves to the next state s' according to the transition probabilities $P(.|s, i, j)$. Thus, the immediate payoffs to the players and the state transition depend on the actions of both the players. There are two different types of stochastic games based on the overall total returns or payoffs received by the players. In first type called infinite-horizon discounted stochastic games, the play begins at state s_0 and proceeds forever. If a player receives payoffs of $r_0, r_1, r_2, \ldots$ as the game progresses through the stages, the expected payoff obtained by the player for the game is given by $r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$, where $0 \leq \gamma < 1$ is the discount factor. In the second type, called finite-horizon undiscounted stochastic game, the play begins at initial state s_0 and proceeds for exactly T steps. If a player receives payoffs of $r_0, r_1, r_2, \ldots, r_{T-1}$, the total payoff for the player is given by $(1/T)(r_0 + r_1 + \ldots + r_{T-1})$. In this work, we have used finite-horizon undiscounted stochastic games since we want the game to stop after T steps. The goal of each player in a stochastic game is to maximize/minimize their expected total payoffs from the designated starting state.

The policy of a player in stochastic game is defined as the mapping $\pi(s)$ from state $s \in S$ to the mixed strategies played by the player at the matrix game during the state s, spanning over all the states of the game. A time-dependent policy $\pi(s,t)$ allows the mixed strategy chosen by the player to depend on the number of remaining steps t in a T-step game. In time-dependent policies, the players gain no advantage by considering the history of the play. If π_1 and π_2 are the policies in a matrix game G with designated start state s_0, $G_k(T, s_0, \pi_1, \pi_2)$, $k \in \{1,2\}$ denotes the expected T-step average return. $\pi = (\pi_1, \pi_2)$ represents the strategy profile of both the players and the expected $T-stage$ payoff to player k, $k \in \{1,2\}$, is given by $\Phi_k^T(\pi)(s) = E_s^\pi \left(\sum_{n=1}^T r_k(s, \pi_k)\right)$, where, E_s^π is the expectation operator with respect to the transition probability P_s^π of the strategies of the players. The average payoff per unit time for player k is defined as

$$\Phi_k(\pi)(s) = limsup \frac{1}{T} \Phi_k^T(\pi)(s).$$

A strategy profile $\pi^* = (\pi_1^*, \ldots, \pi_N^*)$ is called a Nash equilibrium for the average payoff stochastic game if no unilateral deviations from it are profitable. Mathematically, it can be represented as: for each $s \in S$,

$$\Phi_k(\pi^*)(s) \geq \Phi_k(\pi_{-k}^*, \pi_k)(s), \quad \forall k, \pi_k$$

where, (π_{-k}^*, π_k) denotes the strategy profile obtained from π^* by replacing π_k^* with π_k. A matrix game at any given state in a stochastic game may have many Nash equilibria, and hence, there will be exponentially many Nash equilibria in the policy space of the stochastic game [53]. It has been shown in [54] that there exists no polynomial time algorithm to compute an exact Nash equilibrium in a 2-player nonzero sum stochastic games. Hence, we resort to approximate method of finding the Nash equilibrium. A Nash selection function is generally used to convert the local decisions at each state into a global Nash or a near-Nash policy. For any matrix game (M_1, M_2), a Nash selection function f returns a pair of mixed strategies $f(M_1, M_2) = (\alpha, \beta)$ that is a Nash pair for (M_1, M_2). In this case, the

Table 2.2. Notations and Terminology

R_i	resistance per unit length of the interconnect line i
R_{a1}	resistance of first aggressor gate a_1 with min size
R_{a2}	resistance of second aggressor gate a_2 with min size
R_{vd}	resistance of min sized victim driver
U_i	self inductance per unit length of interconnect line i
C_i	self capacitance per unit length of interconnect line i
C_{il}	mutual capacitance per unit length of overlap between interconnect line i and its immediate left neighbor
C_{ir}	mutual capacitance per unit length of overlap between interconnect line i and its immediate right neighbor
C_l	self capacitance of left neighbor line per unit length
C_r	self capacitance of right neighbor line per unit length
C_{vd}	output capacitance of minimum sized victim driver
C_{a1}	output capacitance of minimum sized first aggressor gate a_1
C_{a2}	output capacitance of minimum sized second aggressor gate a_2
C_{mi1}	mutual capacitance of the net i with its first aggressor net
C_{mi2}	mutual capacitance of the net i with its second aggressor net
L	length of the given interconnect line i
W_i	width of the given interconnect line i
W_l	width of the left neighboring interconnect line
W_r	width of the right neighboring interconnect line
S_l	spacing between the given net i and its immediate left neighbor
S_r	spacing between the given net i and its immediate right neighbor
g	gate size of the given driver with respect to its min sized driver
Z_L	load impedance of the given interconnect line i
C_L	load capacitance of the given interconnect line i
T	thickness of the given interconnect line i
H	height of the given interconnect line i from the dielectric
V_i	propagation velocity of the given interconnect line i

payoff to the player 1 is given by $v_f^1(M_1, M_2) \doteq M_1(f(M_1, M_2))$ and the payoff to player 2 is given by $v_f^2(M_1, M_2) \doteq M_2(f(M_1, M_2))$. In other words, a Nash selection function is an arbitrary function used to make choices of how to behave in an isolated matrix game.

2.5 Interconnect Models

In this section, we discuss the interconnect delay, delay uncertainty and crosstalk noise models used in most part of this research. Since transmission line models are more accurate than lumped models in modeling interconnect wires in deep submicron designs (as pointed out in [26, 55]), they are adapted in this work. The notations and terminology used in this dissertation are given in Table 2.2..

2.5.1 Interconnect Delay Models

An analytical equation for interconnect delay of a net is derived based on transmission line analysis in [26]. The propagation delay $D_i(g_{i-1},\, g_i)$ denotes the delay from gate g_{i-1} to gate g_i. The interconnect delay expression is reproduced below in terms of the notations used in this dissertation.

$$D_i = \frac{L}{V_i} + \eta_i \left(R_d + \frac{\sqrt{U_i}}{W_i \sqrt{C_i}} \right) C_L \tag{2.1}$$

where,

$$\eta_i = \frac{\ln 2 \left(e^{\theta_i} + 2\theta_i \left(e^{\theta_i} - 1 \right) \right)}{2}$$

$$\theta_i = \frac{RL\sqrt{C_i}}{2\sqrt{U_i}}, \qquad V_i = \frac{1}{\sqrt{U_i C_i}}$$

Equation 2.1 gives the propagation delay for a single interconnect wire and hence, does not consider the coupling effects due to neighboring wires. We have extended this analytical model to incorporate the coupling effects due to cross capacitance, by replacing self capacitance C_i with total capacitance C_{tot_i}. When performing wire sizing, the coupling effects due to the immediate left and right neighbors have to be considered for the reasons indicated in Section 3.3. Referring to the model developed in [26], the left and right mutual capacitances act in parallel with self capacitance. Hence, while performing wire sizing, the total capacitance is given as $C_{tot_i} = C_i + C_{il} + C_{ir}$. When performing gate sizing, the coupling effects due to the strongest and the second strongest aggressors have to be considered for the reasons indicated in Section 5.3.1. Referring to the model developed in [26], the mutual capacitances due to the aggressors act in parallel with self capacitance for the given interconnect wire. Hence, while performing gate sizing, the total capacitance of the interconnect wire is given as $C_{tot_i} = C_i + C_{mi1} + C_{mi2}$. Combining both wire and gate sizing scenarios, the total capacitance C_{tot_i} can be represented as given in Equation 2.2. Hence, the extended interconnect delay equation which considers the effects of coupling capacitances is given by Equation 2.3.

$$C_{tot_i} = \begin{cases} C_i + C_{il} + C_{ir} & \text{for wire sizing,} \\ \\ C_i + C_{mi1} + C_{mi2} & \text{for gate sizing.} \end{cases} \tag{2.2}$$

$$D_i = \frac{L}{V_i} + \eta_i \left(R_d + \frac{\sqrt{U_i}}{W_i \sqrt{C_{tot_i}}} \right) C_L \tag{2.3}$$

where,

$$\eta_i = \frac{\ln 2 \left(e^{\theta_i} + 2\theta_i \left(e^{\theta_i} - 1 \right) \right)}{2}, \qquad \theta_i = \frac{R_i \sqrt{C_{tot_i}}}{2\sqrt{U_i}},$$

$$V_i = \frac{1}{\sqrt{U_i C_{tot_i}}} \qquad C_{tot_i}\text{is given by Equation 2.2}$$

When performing wire sizing, the effects of parasitic capacitances and resistances have to be captured in terms of the wire sizes and spacings of the neighboring net segments. The analytical expressions for the self capacitance and mutual capacitances are derived in terms of its wire widths and spacings in [56] and [57] respectively. These equations are reproduced below in terms of our model parameters. The self capacitance C_i is given by the Equation 2.4. The mutual capacitance between the given net and its immediate left neighbor is given by Equation 2.5. The mutual capacitance between the given net and its immediate right neighbor can be obtained by replacing the values of width and spacing in Equation 2.5 with the corresponding values of right neighbor.

$$C_i = \epsilon_r \left[10.166 \left(\tfrac{W_i}{H} \right) + 24.752 \left(\tfrac{T}{H} \right)^{0.222} \right] \; pF/m \tag{2.4}$$

$$C_{il} = \frac{55.6\epsilon_r}{ln\left[\pi^2 S_l^2 \left(\frac{1}{W_l + T} \right) \left(\frac{1}{W_i + T} \right) \right]} \; pF/m \tag{2.5}$$

When performing gate sizing, the values of parasitics are treated as constants since they depend only on wire sizes and spacings, and not on gate sizes. Hence, the values of the wire resistance, area and coupling capacitances, and inductance are extracted from the Standard Parasitic Exchange Format (SPEF) netlist generated after detailed routing of the design. These values are used directly in the Equations 2.3 and 2.2.

2.5.2 Delay Uncertainty Models

Delay uncertainty is defined as the deviations or the rate of change of propagation delay due to gate size changes as an impact of process variations. Hence, the analytical equation for delay uncertainty can be obtained by differentiating the propagation delay given by Equation 2.3 with respect to the gate sizes [44]. The delay uncertainty $DU_i(gate_i, gate_{i+1})$ from the gate, $gate_i$ to the next gate, $gate_{i+1}$, due to process variations is given by equation 2.6. The value of C_{tot_i} is given according to the Equation 2.2.

$$DU_i = \eta_i \left(\frac{R_{vd}(g_i - g_{i+1})}{g_i^2} + \frac{\sqrt{U_i}}{W_i \sqrt{C_{tot_i}}} \right) C_L \tag{2.6}$$

2.5.3 Crosstalk Noise Models

When performing wire sizing, the crosstalk noise has to be calculated in terms of the coupling effects (expressed as a function of wire sizes) on the given net due to its immediate left and right neighbors separately. After calculating these values separately, the crosstalk noise on a given net is given by the superposition theorem in terms of the coupling effects due to its left and right neighbors. For determining the crosstalk voltage on a net due to its left neighbor, the given net is connected to ground at its source end and terminated with a load capacitance of C_L at its terminal end. The left neighbor is considered to be driven by a unit step voltage at its source end and terminated with a load capacitance of C_{Ll} at its terminal end. Hence, the crosstalk voltage due to left neighbor can be defined as the voltage $V_l(t)$ induced across the load C_L of the net under consideration. It has been shown in [55] that the amplitude of crosstalk voltage at time t is given by the Equation 2.7.

$$V_l(t) = \frac{1}{2} \left[\exp\left(-\frac{t}{\tau_1}\right) - \exp\left(-\frac{t}{\tau_2}\right) \right] \tag{2.7}$$

where,

$$\tau_1 = R\left(C_i + C_L\right), and\ \tau_2 = R\left(2C_{il} + C_i + C_L\right)$$

Using the theory of maxima and minima of differential calculus, it can be shown that the maximum value of the crosstalk voltage is given by Equation 2.8.

$$\begin{aligned} V_l^{max} &= \tfrac{1}{2}\left[\exp\left[\left(\tfrac{N_{cl}-1}{2N_{cl}}\right)\ln\left(\tfrac{1+N_{cl}}{1-N_{cl}}\right)\right]\right] \\ &-\tfrac{1}{2}\left[\exp\left[\left(-\tfrac{N_{cl}+1}{2N_{cl}}\right)\ln\left(\tfrac{1+N_{cl}}{1-N_{cl}}\right)\right]\right] \end{aligned} \tag{2.8}$$

where, the capacitance coupling coefficient N_{cl} is given by $N_{cl} = C_{il}/(C_i + C_{il} + C_L)$. Similarly, the maximum crosstalk noise V_r^{max} induced across the load C_L of the net under consideration due to its right neighbor is given by Equation 2.9.

$$\begin{aligned} V_r^{max} &= \tfrac{1}{2}\left[\exp\left[\left(\tfrac{N_{cr}-1}{2N_{cr}}\right)\ln\left(\tfrac{1+N_{cr}}{1-N_{cr}}\right)\right]\right] \\ &-\tfrac{1}{2}\left[\exp\left[\left(-\tfrac{N_{cr}+1}{2N_{cr}}\right)\ln\left(\tfrac{1+N_{cr}}{1-N_{cr}}\right)\right]\right] \end{aligned} \tag{2.9}$$

where, the capacitance coupling coefficient N_{cr} is given by $N_{cr} = C_{ir}/(C_i + C_{ir} + C_L)$. The total crosstalk noise on the given interconnect is calculated by applying the superposition theorem for voltages V_l^{max} and V_r^{max}, defined in Equations 2.8 and 2.9 respectively.

When performing gate sizing, the crosstalk noise has to be expressed in terms of the gate sizes and input resistances of the victim and aggressor drivers. An analytical expression for maximum crosstalk noise with victim driver modeled as an effective resistance and aggressor driver modeled as a voltage source connected to its gate resistance is derived in [58]. This equation is reproduced here in terms of our notations. V_x^{max} is the peak value of crosstalk noise between a victim net and its two aggressors.

$$V_x^{max} = \frac{R_v C_c V_{dd}}{\tau_0 t_r}\left(\tau_1 Y_1 \left(\frac{Y_1}{Y_2}\right)^{\frac{\tau_2}{\tau_1-\tau_2}} - \tau_2 Y_2 \left(\frac{Y_1}{Y_2}\right)^{\frac{\tau_1}{\tau_1-\tau_2}}\right) \tag{2.10}$$

where,

$$Y_1 = \exp\left(\frac{-t_r}{\tau_1}\right) - 1, \quad Y_2 = \exp\left(\frac{-t_r}{\tau_2}\right) - 1,$$

$$\begin{aligned} \tau_0 = \{&[R_a(C_a + C_c) + R_v(C_v + C_c)]^2 \\ &-4R_vR_a(C_vC_c + C_vC_a + C_aC_c)\}^{\frac{1}{2}} \end{aligned}$$

$$\tau_1 = \frac{2R_vR_a(C_vC_c + C_vC_a + C_aC_c)}{R_a(C_a + C_c) + R_v(C_v + C_c) + \tau_0},$$

$$\tau_2 = \frac{2R_vR_a(C_vC_c + C_vC_a + C_aC_c)}{R_a(C_a + C_c) + R_v(C_v + C_c) - \tau_0},$$

$$R_a = \frac{R_{a1}R_{a2}}{g_{a2}R_{a1} + g_{a1}R_{a2}}, \qquad R_v = R_i + g_{vd}R_{vd}$$

$$C_a = g_{a1}C_{a1} + g_{a2}C_{a2}, \qquad C_c = C_{mi1} + C_{mi2}$$

CHAPTER 3

WIRE SIZING

In this chapter, we focus on the problem of post layout wire sizing to minimize the DSM effects of interconnects. Specifically, we develop a game theoretic framework and multi-metric optimization algorithms for the simultaneous optimization of (i) interconnect delay and crosstalk noise, and (ii) interconnect delay, power and crosstalk noise, during wire sizing. We formulate the wire sizing optimization problem as a normal form game model and solve it using Nash equilibrium theory. The nets connecting the driving cell and the driven cell are divided into net segments. The net segments within a channel are modeled as players and the range of possible wire sizes forms the set of strategies. The payoff function is modeled (i) as the geometric mean of interconnect delay and crosstalk noise in the case of first formulation, and (ii) as the weighted-sum of interconnect delay, power and crosstalk noise in the second formulation. The net segments are optimized from the ones closest to the driven cell towards the ones at the driving cell. The complete information about the coupling effects among the nets is extracted after the detailed routing phase. The time and space complexities of the proposed wire sizing formulations are linear in terms of the number of net segments. We also provide a mathematical proof of existence for Nash equilibrium solution for the proposed wire sizing formulation.

3.1 Problem Definition

The problem of post layout wire sizing can be defined as finding the optimal wire widths such that interconnect effects are minimized under the given area constraints and without the need for rerouting any of the nets in the design. The parasitic resistance and capacitance of interconnect wires are highly dependent on the wire widths. The coupling capacitance is responsible for the majority of the deep submicron effects. Hence, it is important to extract the coupling capacitance of nets with high accuracy. The coupling capacitance of a net depends on its wire size, the length of overlap and spacing between adjacent nets. This information can be efficiently extracted at post-routing phase. In this work, we have modeled the problem of wire sizing such that it does not require re-routing and does not incur area overhead.

3.2 Motivation for Wire Sizing Problem

The problem of wire sizing has been addressed at logic level or pre layout level by many researchers in the recent past. But, this problem has not been addressed at post layout level before. After the design is routed, the locations and orientations of the transistors and interconnect wires in the design are fixed. The application of optimization methods like buffer insertion, wire shielding at post-route stage would result in area overhead and can lead to rerouting of the design. Re-routing of the design is time-consuming and costly to be performed repeatedly. Typically, when a design is routed, the channels have "unused" tracks which remain as white spaces and go through the fabrication process as wasted resource. Wire sizing can effectively make use of these "unused" tracks available throughout the design, in optimizing the design parameters. If the wire sizing problem is modeled properly, it is possible to achieve optimization without the need for re-routing or additional area overhead. We show that wire sizing could be powerful and effective in making use of the unused routing resources to optimize design parameters at post-route stage.

In [59], it has been shown that wire tapering is not required and uniform wire sizing is sufficient to gain the benefits of delay reduction due to wire sizing. Also, it is pointed out that wire size optimization is not widely used due to the lack of integrated wire sizing design framework. Following this, we divide the nets into segments according to channels and perform uniform wire sizing for each net segment. The works on wire sizing reported in [19, 21, 23, 22] use analytical expressions and the works in [24, 20] use non-linear formulations while targeting for delay optimization. These models do not consider the routing congestion and the net positions, and hence, result in unconstrained wire sizes, which cannot be applied directly for sizing the nets of a routed design. The use of these approaches at post-route level will result in DRC violations, and requires rerouting to fix them. Hence, there is a need to develop a new methodology which integrates in the current design flow and determines the wire sizes within the limits of DRC rules, thereby avoiding the need for re-routing. In this work, we develop a complete design framework capable of performing simultaneous optimization of interconnect delay, power and crosstalk noise through wire sizing at post-route level, satisfying the above requirements.

The wire size of a net will affect the sizes of the neighboring nets resulting in conflicting objectives. As the wire size of a net increases, the interconnect delay decreases and the coupling capacitance, hence crosstalk noise and interconnect power increases (convex payoff function). This suits the modeling of the problem using game theory with the possible wire sizes as strategies and the net segments as players who collectively work towards the global objective of optimizing the interconnect delay, power and crosstalk modeled as the payoff function. Traditionally, this problem is modeled using crosstalk noise as the objective function, while maintaining interconnect delay as a constraint or vice versa. However, game theoretic formulation and Nash equilibrium solution allow the simultaneous optimization of multiple metrics with conflicting objectives. Since, interconnect delay, power and crosstalk noise within a circuit are conflicting in nature, the proposed approach is beneficial. Also, in game models, it is possible to have individual payoff functions which can better capture the coupling effects of individual net segments. The performance of the proposed algorithm is compared with that of simulated annealing and genetic search in order to illustrate the effectiveness of game theoretic solutions for problems with conflicting objectives. It is shown in Section 3.9 that the proposed approach yields better results than simulated annealing and genetic search under the assumptions of same models, setup, parameters and objective function for their implementation.

3.3 Simultaneous Optimization of Delay and Noise

We use a place and route tool to perform the placement and routing of the given gate-level netlist. The global grids of the router are used to partition the complete routing area into distinct rectangular sections called channels. The channel boundaries are used in dividing the nets into various net segments. The following information is extracted from the routed design to calculate the wire resistance, wire capacitance and coupling capacitances accurately:

(i) Net segments belonging to each net

(ii) Channel numbers corresponding to the net segments

(iii) Track numbers in the channel

(iv) Wire length of the net segments in a channel

(v) Starting position of the net segments in a channel

(vi) Metal layers to which the net segments belong

(vii) Direction of the net segment

The minimum wire size of any net segment is fixed based on the minimum wire size design rule requirement of the process technology. The maximum wire size for a net segment is determined from the track distance between its immediate adjacent nets and the minimum edge-to-edge spacing requirements. The range between minimum and maximum wire sizes for each net segment can be treated as its possible wire sizes without violating the process design rules. The range for each net segment is divided into a discrete set of values with equal step sizes. The number of entries in the discrete set are different for each net segment as it depends on its location and its immediate neighbors. The discrete set of allowable wire sizes for a net segment is modeled as its strategy set without violating the design rules.

Algorithm 3.1. Wire Sizing Algorithm for Simultaneous Optimization of Interconnect Delay and Crosstalk Noise

```
Input: Placed and routed design
Output: Optimized wire sizes
  Algorithm:
  extract the net information
  organize the nets into channels and tracks
  identify terminal net segments
  for all layers do
    for all channels do
      initialize loads();
      initialize scores();
      determine strategies();
      mark the channel as un-played
    end for
  end for
  select a channel $i$ with lowest score value
  while there exists an un-played channel do
    calculate mutual-capacitance();
    calculate wire-capacitance();
    calculate wire-resistance();
    for all net segments $j \in$ channel $i$ do
      create a 3-player game with $j$ and its left and right neighbors
      cost-matrix ← payoff(three players, strategies)
        % for payoff function, see Algorithm 3.2.
      optimized-width ← nash-solution(three players, payoffs)
        % for Nash equilibrium solution, see Algorithm 3.3.
    end for
    update loads();
    update scores();
    mark the channel as played
    select the a new channel with lowest score value
  end while
  return: optimized widths of all net segments
```

A game is modeled for each individual channel. The channels located on different layers are considered separately as they consist of different net segments. For a given channel, its net segments are modeled as the players of the game. The coupling effect on a net segment depends on all the net segments adjacent to it. As the distance between the net segments increases, the coupling capacitance between them decreases rapidly thereby reducing the coupling effects due to each other. As pointed out in [60], in the context of wire sizing, it is sufficient to consider the coupling effects due to its immediate neighbors for two reasons: (i) The coupling effects of other neighboring nets are minimal when compared to immediate neighbors due to their increased distance from the given net. (ii) The immediate neighbors acts as shields to the given net from the other neighboring nets. Hence, in this work, we consider the coupling effects due to its immediate left and right neighbors for a given net segment.

Algorithm 3.2. Algorithm for Payoff Matrix Calculation for Simultaneous Optimization of Interconnect Delay and Crosstalk Noise

```
Input: Number of Players $N$, Strategy set $S$
Output: Payoff matrix
  Algorithm:
  for all players $i \in 1$ to $N$ do
    for all strategy combinations $S^j = \{s_1^j, \ldots, s_N^j\}$, where $(s_1^j \in S_1), \ldots, (s_N^j \in S_N)$ do
      calculate the delay using Equation 2.3
      normalize the delay w.r.t first strategy combination
      calculate the crosstalk noise using Equations 2.8 and 2.9
      normalize the noise w.r.t first strategy combination
      $P[i, S^j] \leftarrow$ Geometric mean of normalized noise and delay
    end for
  end for
  return: payoff matrix $P$ $\forall$ strategy combinations
```

The payoff function tries to capture the interaction between the neighboring net segments (modeled as the players of the game) in the channel. For each net segment in a given channel, its delay D and maximum crosstalk noise N are calculated by using the Equations 2.3, 2.8, and 2.9. These values are calculated for all strategies of the given net segment by considering the strategies of its immediate left and right neighbors. The delay and crosstalk noise values obtained for a net segment are then normalized with respect to the corresponding first strategy. The normalization is performed to transform the delay and crosstalk noise values into dimension-less quantities so that they be easily correlated with each other. The payoff function is modeled as the geometric mean of normalized delay and normalized crosstalk noise values for each strategy of the net segment. We have chosen geometric mean so as to give equal weights to both crosstalk noise and interconnect delay components during their optimization.

Algorithm 3.3. Algorithm for Nash Equilibrium Solution

```
Input: Number of players $N$, Payoff matrix $P$
Output: Nash solution
  Algorithm:
  for all players $i$ do
    for all payoffs of player $i$ do
      find $s_i^*$ such that,
      $P_i(s_1^*, \ldots, s_i^*, \ldots, s_N^*) \geq P_i(s_1^*, \ldots, s_i, \ldots, s_N^*)$
      $s_i^*$ is the Nash strategy for player $i$
    end for
  end for
  Nash-solution $S^* = \{s_1^*, \ldots, s_N^*\}$
    % set of optimized strategies for all $N$ players
  return: Nash solution $S^*$
```

We have used normal form formulation to mathematically represent and solve the game. Normal form game representation suits formulations well because it emphasizes mainly the competition between the players participating in the game. In addition, normal form games can be easily modeled and implemented as they require only a payoff matrix and the corresponding strategy set to reach the game's Nash equilibrium. Mathematically, a normal form game consists of a set of N players labeled $1, 2, \ldots, N$, such that each player i has

(i) a choice set S_i called strategy set of player i; its elements are called strategies.

(ii) a payoff function $P_i : S_1 \times S_2 \times .... \times S_N \rightarrow \Re$, assigned to each strategy chosen by the player i with respect to other players.

In a normal form game, all the players play simultaneously without any knowledge about other players' play. In other words, the players simultaneously choose a strategy $s_i \in S_i$ such that their respective payoff is maximized or minimized with respect to the payoffs of the other players. The equilibrium of the game is computed by using the Nash equilibrium condition. Consider a channel consisting of N net segments. The wire size of any net segment in the channel is influenced only by its immediate left and right neighbors. Therefore, the payoff function of any net segment in the channel depends only on two other players (left and right immediate neighbors) rather than on all the N-1 players. Thus, for each channel, instead of having a single game with N-players, we divide the game into N sub-games with each sub-game involving 3-players - the given net segment, its left neighbor and its right neighbor.

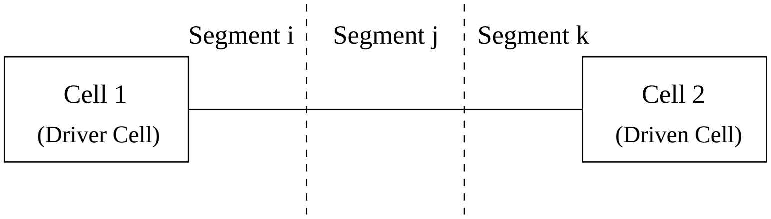


Figure 3.1. An Example Scenario

The interconnect delay calculations for each net segment require its values of wire and load capacitance, in addition to other parameters defined in Equation 2.3. As an example, consider a net connected between cell 1 and cell 2 with cell 1 driving the net and cell 2 receiving, as shown in Figure 3.1.. In this example, the net is divided into three segments just as an example for illustration. The load capacitance of segment k is the input capacitance of the cell 2, which is known. The cell 2 and segment k act as loads for segment j. The wire capacitance of segment k depends on its wire width. Hence, in order to calculate the load capacitance of segment j, the wire width of segment k has to be optimized, requiring segment k to play the game before segment j. In general, the load capacitance of a net segment can be calculated only when its down-stream wire segments are optimized. Hence an ordering for channels to play has to be defined which satisfies the load capacitance dependency. A score, defined as the difference between the total net segments and the number of terminal nets belonging to a channel, is used for ordering the channels. The ordering of channels aids in considering the effects of wire sizes of down-stream net segments. Even though the game is played for a segment, the load capacitance takes into account the effects of its complete net. Thus the resulting solution is not a local solution which is confined to a segment of the net.

A channel with lowest score is selected to play the game with its non-terminal net segments assigned with a default load capacitance. Nash equilibrium is evaluated for the lowest score channel and its Nash widths are used to update the load capacitances of the net segments belonging to its adjacent channels. The scores of only the neighboring channels have to be updated to reflect the net segments with known load values as terminal net segments. Hence, when a channel is played out, the load and score values of a maximum of six adjacent channels have to be updated to reflect the optimal widths for the net segments resulted from the played channel. Again, a channel with lowest score is selected to play the next game and this process is repeated until all the channels are played out. Algorithm 3.1. shows the pseudo-code of the complete wire sizing algorithm for simultaneous optimization of delay and crosstalk noise.

3.4 Simultaneous Optimization of Delay, Power and Noise

In section 3.3, we described a wire sizing methodology for optimizing interconnect delay and crosstalk noise. In this section, we develop a wire sizing methodology for multi-metric optimization of interconnect delay, power and crosstalk noise. Again, the placed and routed design is used as the starting point. The net extraction phase, the strategy generation and the game modeling for this formulation are similar to the one described in section 3.3. The main difference between the two formulations is the modeling of the payoff function. The objective function or the payoff function is a combination of the interconnect delay, power and crosstalk due a net segment. The interconnect power can be modeled as the power dissipated due to the charging and discharging of the capacitance exhibited by the interconnect wires. Mathematically, interconnect power can be represented as

$$P_{inter} = \frac{1}{2}\alpha V_{dd}^2 f_{clk} C_{inter}$$

where, α is the switching activity of the interconnect wire, V_{dd} is the supply voltage and f_{clk} is the clock frequency. The interconnect capacitance, C_{inter}, is the sum of the wire and coupling capacitances. Hence, the power dissipated due to interconnects is given by equation 3.1.

$$P_{inter} = \frac{1}{2}\alpha V_{dd}^2 f_{clk}(C_i + C_{il} + C_{ir}) \tag{3.1}$$

The payoff function or the objective function is modeled as the weighted sum of normalized delay, power and crosstalk noise values for each net segment. For each net segment in the channel, the delay D is calculated using equation 2.3, the crosstalk noise N is calculated as the sum of values given by equations 2.8 and 2.9, and the interconnect power P is given by equation 3.1. These values are calculated for all strategies of the given net segment by considering the strategies of its immediate left and right neighbors. The delay, noise and power values obtained for each net segment are normalized with respect to their first strategy. The weights for delay D, power P and crosstalk noise N can be adjusted by the designer according to the need. In this work, we have chosen 0.33 as the weight for both crosstalk noise and interconnect power, and 0.34 as the weight for interconnect delay. The algorithm for calculating the payoff function is given in Algorithm 3.4..

Algorithm 3.4. Algorithm for Payoff Matrix Calculation for Simultaneous Optimization of Interconnect Delay, Power and Crosstalk Noise

```
Input: Number of Players $N$, Strategy set $S$
Output: Payoff matrix
  Algorithm:
  for all players $i \in 1$ to $N$ do
    for all strategy combinations $S^j = \{s_1^j, \ldots, s_N^j\}$, where $(s_1^j \in S_1), \ldots, (s_N^j \in S_N)$ do
      calculate delay $D$ using Equation 2.3
      calculate crosstalk noise $N$ using Equations 2.8 and 2.9
      calculate power $P$ using Equation 3.1
      $P[i, S^j] \leftarrow$ a $*D$ + b $*P$ + c $*N$
        % a, b, c are the weights of Delay, Power and Noise respectively
    end for
  end for
  return: payoff matrix $P$ $\forall$ strategy combinations
```

3.5 Time and Space Complexity of Proposed Wire Sizing Algorithms

The worst case time complexity of evaluating Nash equilibrium for a general M-player game with S strategies for each player is given as $O(M * S^M)$ [52]. Referring to Section 3, we have modeled the problem of wire sizing as a set of 3-player games for each net segment. The incremental step size between two consecutive wire sizes for any net segment is kept constant. Hence, the number of strategies for each player depends on its range of possible wire sizes, which is different from player to player. In this work, we have chosen the step size such that the number of strategies for any net segment is less than five. Consider a channel with N net segments. Each net segment in the channel will form a 3-player game with its left and right neighbors. Hence, the complexity of calculating Nash equilibrium for a given channel with N net segments is given as $O(N * 5^3) \approx O(N)$. The Nash equilibrium chooses optimal wire sizes for the players considering each game individually. But, a player participates in three different games formed for itself, its right neighbor and its left neighbor. We noticed from our experiments that the widths resulting from the three games are equal for around 70% of net segments. In case of different widths, maximal likelihood Nash width is assigned to the net segment. Considering all the channels and the layers in a given design, the overall time complexity of proposed algorithm is given as

$$O\left(L * \sum_{i=1}^{C} N_i\right) = O\left(\sum_{\forall i \in nets} n_i\right)$$

where L is the total number of layers, C is the number of channels in a layer, N_i is the number of net segments belonging to channel i and n_i is the number of net segments in which a net i is divided. Hence, the time complexity of the proposed algorithm is *linear* in terms of total net segments in the design.

The space complexity of the proposed algorithm is dependent entirely on the number of net segments in the design and the payoff matrix. The space complexity of the payoff matrix depends on the number of strategies for each player in the game. As the games are played sequentially, the total space required by all games put together is equal to the space complexity of a game involving players with maximum number of strategies. Hence, mathematically, the space complexity is given as $O(S_1 * S_2 * S_3) \leq O(5 * 5 * 5)$, where S_1, S_2, and S_3 are the strategy sets of 3-player game involving the players with maximum strategies. Hence, the space complexity of the proposed algorithm is given as

$$O\left(\sum_{\forall i \in nets} n_i + 5*5*5\right) \approx O\left(\sum_{\forall i \in nets} n_i\right)$$

3.6 Proof of Existence of Nash Equilibrium Solution for the Wire Sizing Formulation

In this section, we provide the proof of existence of Nash equilibrium in the case of wire sizing problem for simultaneous optimization of interconnect delay and crosstalk noise. As the wire size of a net increases, the interconnect delay decreases and the coupling capacitance increases resulting in a convex payoff function. Let $G = \{S_1, \ldots, S_n; f_1, \ldots, f_n\}$ be a game with each player $i \in N$ having a strategy set S_i containing its possible wire sizes and its payoff given by f_i. We have modeled the strategy set S_i for each player as a non-empty, compact set of a finite dimensional Euclidean space. Because of the convex nature of the interconnect delay and crosstalk noise, the modeled payoff function f_i becomes upper semicontinuous on $S = \Pi_{i=1}^{N} S_i$ and for any fixed $u_i \in S_i$, the function $f_i(u_i, .)$ is a lower semicontinuous on $S_{(-i)}$ [16]. For any $u \in S$, the best reply or the expected payoff $B_i(u)$ is also convex. According to Kakutani's fixed point theorem [61], the game G has at least one Nash equilibrium point if the graph

$$G_B = \{(x, y) : x \in S, y \in B(x)\}$$

is closed.

Lets assume that its not closed. Then, $\exists (x^0, y^0) \notin G_B$, such that every neighborhood (in $S \times S$) of (x^0, y^0) contains a point of G_B.

$\because x^0$ is a wire size, it has to be one of those from the set of possible wire sizes for the given player in order to satisfy the DRC rules of the used process technology.

$\therefore x^0 \in S \Rightarrow y^0 \notin B(x^0)$

In other words, for at least one net segment playing the game (say segment 1), there is an $y_1^1 \in S_1$ such that

$$f_1(y_1^1, x_2^0, \ldots x_n^0) \geq f_1(y_1^0, x_2^0, \ldots x_n^0) \tag{3.2}$$

Let F be a function such that $F : S^2 \rightarrow \Re$ and given as

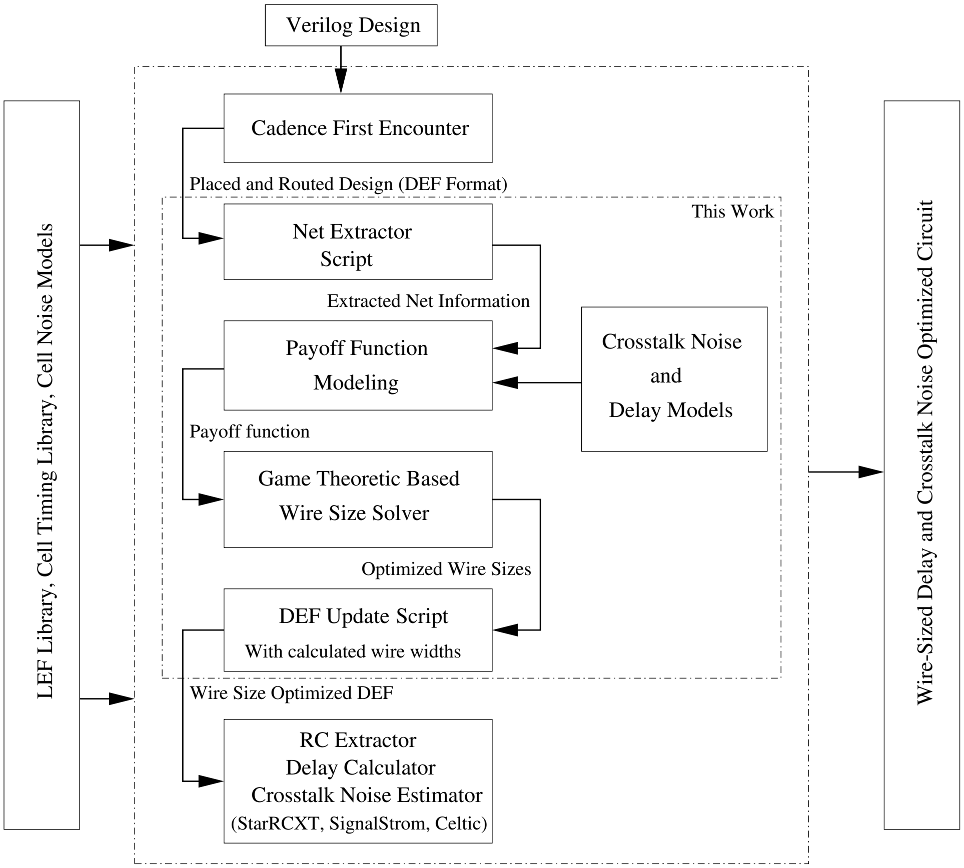
$$F(x, y) = f_1(y_1^1, x_2, \ldots x_n) - f_1(y_1, x_2, \ldots x_n)$$

Since f_i is upper semicontinuous on S and $f_i(u_i, .)$ is lower semicontinuous on S_{-i}, F is lower semicontinuous and $C = \{(x, y) \in S^2 : F(x, y) \leq 0\}$ is closed. Hence, for any $(\bar{x}, \bar{y}) \in G_B, F(\bar{x}, \bar{y}) \leq 0$. But, by Equation 3.2, $F(x^0, y^0) \geq 0$, contradicting the closedness of C. Thus, there is a point $s^* \in S$ such that $s^* \in B(s^*)$, which is a Nash equilibrium point.

3.7 Discussion

In this section, we explain the rationale behind the optimization of entire nets of the design rather than only the critical nets. Wire sizing technique can efficiently utilize the “unused” routing resources to minimize the design parameters of a routed design and hence, it is advantageous to be applied at post-route phase of the design. In the context of wire sizing at post-route phase, the maximum size with which a net can be sized is fixed. The sizing of a net has to be performed within this feasible range or else a considerable number of nets have to be re-routed. The re-routing of a design usually requires tremendous amount of time and effort. This is valid for critical nets as well and hence, have to be sized within the routing resources available to it. With the available routing resources, the game theoretic formulation allows a better allocation for the critical nets when compared to its neighbors. This is because the payoff values for critical nets dominates that of its neighbors and hence, Nash equilibrium gives more weight to the critical nets and results in a solution which is in the best interest of both critical nets and their neighbors. The routing resources available at other locations can be better used to optimize the corresponding nets, rather than leaving them unused. Hence, we have planned to optimize all the nets in the design. Also, optimizing all the nets in design will have an advantage of enforcing the timing closure, the signal integrity for all nets and hence, aids in other post-layout optimization techniques. The experimental results validate our claims depicting better critical net savings for our approach when compared to the simulated annealing or genetic search.

3.8 Design Flow



Verilog Design
LEF Library, Cell Timing Library, Cell Noise Models
Cadence First Encounter
Placed and Routed Design (DEF Format)
This Work
Net Extractor Script
Extracted Net Information
Payoff Function Modeling
Crosstalk Noise and Delay Models
Payoff function
Game Theoretic Based Wire Size Solver
Optimized Wire Sizes
DEF Update Script
With calculated wire widths
Wire Size Optimized DEF
RC Extractor
Delay Calculator
Crosstalk Noise Estimator
(StarRCXT, SignalStrom, Celtic)
Wire-Sized Delay and Crosstalk Noise Optimized Circuit

Figure 3.2. Integration of Proposed Wire Sizing Algorithm in the Design Flow

The design flow for obtaining an optimally wire sized circuit from a verilog/VHDL description is shown in Figure 3.2.. The behavioral verilog/VHDL description is synthesized on to a library of standard cells and given as input to the design flow. The standard cells are placed and routed in accordance with the synthesized code using any standard cell place and route tool. We have used the First Encounter™ RTL-to-GDSII tool from Cadence® Design Systems to perform the placement and routing of gate-level RTL design. The net information required for calculations of interconnect delay and crosstalk noise is extracted from the routed design. A gawk script is developed which extracts this information from the exported DEF file of the routed design. The payoff function is used by the game theoretic based wire size solver described in Algorithm 3.1. to minimize the interconnect delay and crosstalk noise of every individual net of the design. The optimized wire sizes resulted from the game theoretic wire size solver are used to update the routed design. We have developed another gawk script which updates the wire sizes of all the nets in the original DEF routed design with their corresponding optimized wire sizes. It should be noted here that the resulting optimized design does not require re-routing as all the sized nets satisfy the design rules of the given process technology.

3.9 Experimental Results

We have implemented the proposed algorithm in C and executed on a UltraSPARC-IIe 650MHz, 512MB Sun Blade 150 system operating on Solaris 2.8 and tested with the ASIC designs from Opencores [62]. A 180nm, 6-Metal standard library is obtained from Crete [63], an educational university campus program developed and maintained by Cadence design systems. The standard cell library contains about 40 logic cells and over 100 I/O cells with the corresponding cell timing and transistor models. ASIC designs, written in behavioral VHDL/Verilog are converted to structural VHDL/Verilog using the standard cells in the library with the help of BuildGates, an RTL synthesis tool of Cadence design systems. We have modified the ASIC designs such that all the blocks in the design are flattened to standard cells in the library without maintaining the hierarchy. The on-chip memory modules are realized as D-flipflop register arrays. The structural VHDL/Verilog design is used as input by Cadence First Encounter to develop the floorplan. We have set the option of row utilization to 95% for all the designs so as to have a compact floorplan. The design is then placed and routed using Amoebaplace and Nanoroute respectively, which are part of Cadence First Encounter tool. The final placed and routed design is then exported in DEF format. The net information is extracted from the DEF file and provided as input to the game theoretic wire size solver. The calculated wire size for each net is used to update the wires in the original DEF file to generate an optimized DEF file. It can be noted that the optimized DEF is created with the help of gawk scripts and is not re-routed. The parasitic resistances and capacitances from both original and optimized DEF files are extracted using StarRCXT from Synopsys Inc. The interconnect delay and crosstalk noise are estimated using Cadence Signalstorm and CelticIC tools respectively.

Several of the pioneering works reported in the literature for the problem of wire size optimization [19, 21, 23, 22, 25, 26], only present results for arbitrary nets and do not consider routing congestion, floorplan compaction, etc, of the specific design or benchmark circuits. *Thus, it is not possible to provide a direct comparison of our results with those works.* To compare our results, we have implemented simulated annealing and genetic search based algorithms and executed on the same Solaris machine with same set of inputs and constraints. The annealing process of simulated annealing approach is determined by experimenting significantly to get the best results and the maximum optimization. The nets are divided into net segments and the set of possible wire sizes for each net segment is calculated as indicated in Section 3.3. In each move of the annealing process, a net segment is randomly selected and its size is assigned from the set of its possible wire sizes. The cost function is defined as the geometric mean of the interconnect delay and the crosstalk noise summed over all the net segments. The initial temperature is determined by finding the average change in the cost for a set of random moves from the starting configuration and selecting the temperature which leads to an accept probability of 0.95. The number of moves per temperature for each design is set to 20 times the number of net segments in the design so as to allow an average of at least 10 to 15 moves for each net segment before settling for its solution. The up-hill moves are accepted with a probability of $e^{-\frac{\delta C}{T}}$, where δC is the change in the cost and T is the current temperature of the iteration. The temperature is cooled at the rate of 0.95.

Table 3.1. Experimental Results for Simultaneous Optimization of Interconnect Delay and Crosstalk Noise During Wire Sizing Using Complex Delay Models

Open core Design	Total Nets	Die Area (mm^2)	Genetic Search*					Simulated Annealing Approach*					Game Theoretic Approach *				
			Run time (mins)	% Delay Savings		% Noise Savings		Run time (mins)	% Delay Savings		% Noise Savings		Run time (mins)	% Delay Savings		% Noise Savings	
				Avg.	Crit.	Avg.	Crit.		Avg.	Crit.	Avg.	Crit.		Avg.	Crit.	Avg.	Crit.
Mult	854	0.199	34.16	2.45	5.32	3.10	4.32	14.32	6.15	8.91	5.12	13.26	1.89	9.87	11.79	12.13	17.23
PCI bus	19520	0.434	183.23	12.79	20.16	13.78	19.96	47.35	10.31	21.22	12.36	23.39	5.23	19.32	34.21	21.42	37.38
Serial ATA	43563	1.624	418.31	17.94	31.02	11.53	25.82	124.83	18.91	28.65	12.41	26.37	11.86	29.87	39.95	20.14	42.15
RISC	61468	2.102	729.47	11.91	17.37	9.19	15.46	188.33	20.39	39.89	16.31	24.25	16.86	25.22	35.21	22.31	29.73
AVR μP	78770	11.103	972.51	10.18	11.25	13.67	22.13	232.67	17.57	21.35	27.67	40.12	21.32	22.45	37.63	31.34	40.31
P16C55 μC	102021	19.984	1301.43	11.64	15.48	25.91	29.18	288.36	17.98	29.47	31.67	39.45	28.98	19.86	27.45	43.29	57.98
T80 μC	157850	30.388	1689.24	13.76	14.11	15.23	18.10	353.25	14.86	15.97	23.39	29.74	39.48	23.78	34.87	33.12	39.89
Average				11.52	16.39	13.63	19.28		15.17	23.64	18.42	28.08		21.48	31.26	26.25	37.81

[*] No area overhead for all three approaches. The percentage values indicated are w.r.t placed and routed design without wire sizing.

[1] *Table Legend: Avg: Average savings of all the nets in the entire design;*
Crit: Savings on the critical path net of the design;
Runtime: indicates the running time of each algorithm.

Table 3.2. Experimental Results for Simultaneous Optimization of Delay, Power and Noise During Wire Sizing Using Complex Delay Models

Open core Design	Total Nets	Die Area (mm^2)	Game Theoretic Approach						
			Run time (mins)	% Delay Savings		% Power Savings		% Noise Savings	
				Avg.	Crit.	Avg.	Crit.	Avg.	Crit.
Mult	854	0.199	2.34	4.31	7.49	5.16	7.44	8.15	13.56
PCI bus	19520	0.434	8.12	10.69	15.40	14.23	17.05	19.52	25.81
Serial ATA	43563	1.624	14.03	17.90	21.45	17.86	23.61	15.85	24.40
RISC	61468	2.102	23.80	19.92	22.31	15.52	19.03	17.64	24.74
AVR μP	78770	11.103	30.05	13.54	19.40	11.74	16.89	18.22	23.01
P16C55 μC	102021	19.984	37.58	14.11	18.58	19.35	27.20	21.65	30.94
T80 μC	157850	30.388	46.35	18.25	25.05	20.05	28.02	23.94	32.69
Average				14.10		14.84		17.85	

The area overhead incurred is zero. The percentage values indicated are w.r.t placed and routed design without wire sizing. *For table legends, please refer to Table 3.1.*

The wire sizing problem for simultaneous interconnect delay and crosstalk noise optimization is also modeled as a genetic search mechanism and solved using GALib [64]. The initial population contains the net segments with their corresponding wire sizes as used in the original unsized design. Each individual in the population called chromosome is represented as a set of three integers indicating the net number, the segment number and the wire size assigned to the segment. The chromosomes evolve through successive iterations called generations. During each generation, the chromosomes are evaluated for their fitness test. We have defined the fitness criterion as the deviation of the crosstalk noise and interconnect delay of each net segment from its worst-case values. The chromosomes with lower values of crosstalk noise and interconnect delay are given higher fitness values. We have used steady-state genetic algorithm available as a part of GALib library to generate overlapping populations which retains its 30% of fittest chromosomes in its new generations. The mutation process for a chromosome is defined to randomly select a wire size from its set of possible wire sizes. The new chromosomes are created using single point crossover and are validated against their set of possible wire sizes. The selection process of chromosome is adopted by the roulette wheel selection approach. We have set the convergence-of-population as the stopping measure for the evolution of generations.

Table 3.1. shows the experimental results for the case of simultaneous optimization of interconnect delay and crosstalk noise. First column indicates the name of the design and the second column indicates its corresponding number of nets. The area indicated in third column is chip area occupied by the core without considering its I/O pins. The fourth, ninth and fourteenth columns indicate the runtime of genetic search, simulated annealing and game theoretic wire size solvers respectively. Columns five and seven indicate the average delay and noise savings for all the nets of the design obtained by the genetic search mechanism. Columns ten and twelve indicate the same for simulated annealing approach, while Columns fifteen and seventeen represents the game theoretic approach. Columns six and eight indicate the critical net delay and noise savings obtained by the genetic search mechanism. Columns eleven and thirteen indicate the same for the simulated annealing approach, and Columns sixteen and eighteen indicate the game theoretic approach. Table 3.2. shows the experimental results for the case of simultaneous optimization of interconnect delay, interconnect power and crosstalk noise.

The experiments were conducted such that the area overhead is zero in all three approaches. The savings obtained in terms of interconnect delay, power and crosstalk noise depend on the factors like floorplan compaction, routing congestion. This is because the routing congestion decides the wire size scaling of the nets routed through that region. It can be noticed that the game theoretic approach yields better savings than genetic search and simulated annealing for all the test case designs. In addition, our algorithm has significantly smaller run times than genetic search or simulated annealing for fairly large-scale designs. Hence our approach is scalable and favorable to handle the complexity of large SOC designs.

3.10 Conclusions

Game theory allows the simultaneous optimization of multiple metrics in the context of conflicting objectives leading to a convex objective function in the problem formulation. This essentially makes it possible to use game theory for simultaneous optimization of interconnect delay and crosstalk noise. Optimizing both interconnect delay and crosstalk noise is extremely critical in deep submicron and nano regime circuits. The use of game theory and Nash equilibrium for the problem of wire sizing to optimize interconnect delay and crosstalk noise is being attempted for the first time. The proposed method results in a linear time algorithm with significantly better results than simulated annealing, making this work an important contribution.

Our intention in this work was to show that wire sizing can be used to achieve simultaneous optimization of interconnect delay and crosstalk noise at post-route stage. We observed that the previous algorithms for wire sizing target for single metric optimization with other parameters as constraints and have not been tested with all the design constraints such as routing congestion, floorplan compaction, position of nets, etc. Further, prior works have not indicated a viable design flow to include wire sizing [59, 25]. It has been pointed out in [59] that wire tapering for the entire net can yield 5% more savings in delay when compared to uniform wire sizing. However, performing uniform wire sizing within a net segment for all the segments of a net can yield significant savings in terms of crosstalk noise and interconnect delay at post-route stage.

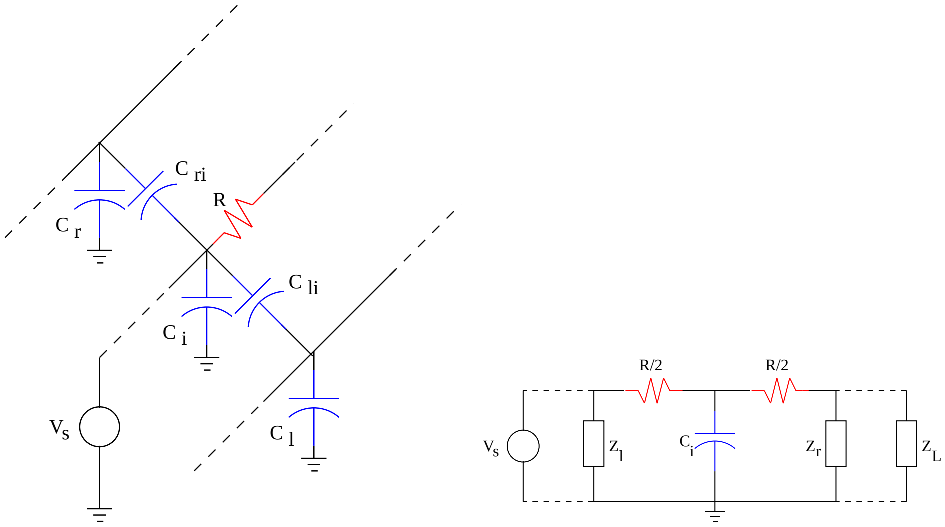
CHAPTER 4

NEW INTERCONNECT MODELS

In the game model developed in Chapter 3, we divided the nets into net segments and modeled the net segments as the players of the game. Even though, a player is modeled as a net segment, the effects of the entire net on its net segment is considered by ordering the channels, which aid in accounting for the downstream effects (please refer to Section 3.3). Hence, the interconnect delay has to be calculated for a net segment rather than the entire net. The lengths of the net segments is sufficiently small and do not result in tree topologies. The equation 2.3 gives the interconnect delay for the entire net and is accurate, but complex to be computed repeatedly for the net segments. Hence, we develop simple, fast but sufficiently accurate interconnect models that can be used for calculating interconnect delays repeatedly in an arrangement of three parallel net segments.

4.1 Fast Transmission Line Models

In this section, we derive new, simple and fast models based on transmission line theory. The net segment is modeled as a transmission line driven by a voltage source and terminated by a load Z_L, as shown in Figure 4.1.(a). We have considered the coupling effects due to the immediate left and right net segments while developing the model for interconnect delay through a segment of the given net. The series resistance of the neighboring net segments is not considered since it does not affect the propagation characteristics along the given net [65]. Figure 4.1.(b) shows the equivalent model with uncoupled capacitances for one section of interconnect line arrangement given in Figure 4.1.(a). The elements shown in Figure 4.1. are defined per unit length of the interconnect line. Referring to Figure 4.1.(b), the series impedance per unit length Z_s and parallel impedance per unit length Z_p are given by Equations 4.1 and 4.2 respectively. The impedances Z_l and Z_r represent the coupling effects due to left and right neighbors considering the propagation along the net segment.



(a) Interconnect modeled as transmission line

(b) Uncoupled equivalent of one section of interconnect

Figure 4.1. Interconnect Model (a) Modeled as a Transmission Line (b) Uncoupled Equivalent of One Section of the Interconnect

$$Z_s = R \tag{4.1}$$

$$\frac{1}{Z_p} = \frac{1}{Z_l} + \frac{1}{Z_r} + sC_i = \left(Y_l + Y_r + C_i\right) s \tag{4.2}$$

$$\text{where, } Z_l = \frac{C_{il} + C_l}{sC_{il}C_l} = \frac{1}{sY_l}, \quad Z_r = \frac{C_{ir} + C_r}{sC_{ir}C_r} = \frac{1}{sY_r}$$

The propagation constant γ and the characteristic impedance Z_0 of the transmission line model are given by the Equations 4.3 and 4.4 respectively.

$$\gamma = \sqrt{\frac{Z_s}{Z_p}} = \sqrt{Rs\left(Y_l + Y_r + C_i\right)} \tag{4.3}$$

$$Z_0 = \sqrt{Z_s Z_p} = \sqrt{\frac{R}{s\left(Y_l + Y_r + C_i\right)}} \tag{4.4}$$

In Laplace transform domain (s-domain), the voltage and current distributions along an interconnect length (denoted by the coordinate z) satisfy the transmission line equations given by 4.5

$$\frac{d^2V}{dz^2} = \gamma^2 V, \; and \; \frac{d^2I}{dz^2} = \gamma^2 I \tag{4.5}$$

The general solutions for the voltage and current satisfying the set of differential equations 4.5 are given by:

$$v_z = Ae^{\gamma z} + Be^{-\gamma z}, \; and \; i_z = \frac{Ae^{\gamma z} - Be^{-\gamma z}}{Z_0} \tag{4.6}$$

where, the constants A and B can be determined by using the two known boundary conditions: (i) the interconnect is driven by voltage source V_s and (ii) the interconnect is terminated by a load Z_L. Quantitatively, they can be represented as:

at source, z = 0 and $v_z = V_s$

at load, z = L and $v_z/i_z = Z_L$, where $Z_L = 1/sC_L$

substituting the above boundary conditions in Equations 4.6, and solving for A and B, we have,

$$A = \frac{V_s\left(Z_L + Z_0\right)e^{-\gamma L}}{e^{\gamma L}\left(Z_L - Z_0\right) + e^{-\gamma L}\left(Z_L + Z_0\right)}$$

$$B = \frac{V_s\left(Z_L - Z_0\right)e^{\gamma L}}{e^{\gamma L}\left(Z_L - Z_0\right) + e^{-\gamma L}\left(Z_L + Z_0\right)}$$

Therefore, the voltage at the load end $(z = L)$ is given by Equation 4.7.

$$V_L = \frac{2V_s Z_L}{e^{\gamma L}\left(Z_L - Z_0\right) + e^{-\gamma L}\left(Z_L + Z_0\right)} \tag{4.7}$$

The steady state voltage at the load of the interconnect line has to reach an ideal value of V_s, the source voltage. The propagation delay in Laplace domain is defined as the time taken for the voltage at the load terminal to reach 50% of its steady state value. At this point, the s-parameter is defined as s = $2\pi/T_{delay}$ [65]. Hence, we have,

$$V_L^{Steady\_State} = \frac{V_s}{2} = \frac{2V_sZ_L}{e^{\gamma L}\left(Z_L - Z_0\right) + e^{-\gamma L}\left(Z_L + Z_0\right)}$$

transforming the above equation, we have

$$\left(e^{\gamma L} + e^{-\gamma L}\right) - \frac{Z_0}{Z_L}\left(e^{\gamma L} - e^{-\gamma L}\right) = 4 \tag{4.8}$$

It should be noted in the above equation that Z_L, Z_0 and γ are functions of T_{delay} in terms of s. Equation 4.8 can be solved for T_{delay} by using the Maclaurin's series expansion. We have approximated the expansion series to second order terms so as to obtain a quadratic equation in terms of T_{delay}, which can be solved easily. Higher order terms of the Maclaurin's series can be included if more accuracy is needed. The resulting model is simple because it ignores wire inductance and limits the number of terms in Maclaurin series.

4.2 Experimental Results

We have performed experiments by using the delay models developed in this chapter to analyze their accuracy when compared to the results provided in Table 3.1. using the complex models. The experiments were performed using the wire sizing methodology for simultaneous optimization of interconnect delay and crosstalk noise developed in Chapter 3. The results incorporating the new models are given in Table 4.1..

Table 4.1. shows the experimental results for game theoretic approach using the delay models defined in Equation 4.8. The first three columns of Table 4.1. represents the same values indicated in the corresponding columns of Table 3.1.. Fourth column indicates the runtime of the game theoretic wire size solver. Columns five and seven indicate the average

Table 4.1. Experimental Results for Game Theoretic Approach Using the Developed Fast Models During Wire Sizing

Opencore Design	Total Nets	Die Area (mm^2)	Run time (mins)	% Delay Savings		% Noise Savings	
				Avg.	Max.	Avg.	Max.
Mult	854	0.199	< 1	8.06	13.32	11.31	16.86
PCI bus	19520	0.434	3.42	17.23	35.68	19.24	36.12
Serial ATA	43563	1.624	8.62	26.58	42.12	16.83	45.33
RISC	61468	2.102	10.68	22.12	44.36	20.46	31.16
AVR μP	78770	11.103	14.23	24.45	38.78	33.26	47.38
P16C55 μC	102021	19.984	19.34	17.68	36.57	40.12	51.46
T80 μC	157850	30.388	26.54	22.45	39.18	32.48	47.19
Average				19.8		24.81	

[*] The results indicated for game theoretic approach has no area overhead. The percentage values indicated are with respect to placed and routed design without wire sizing.

[1] *Table Legend: Avg: Average savings of all the nets in the entire design;*
Crit: Savings on the critical path net of the design;
Runtime: indicates the running time of each algorithm.

interconnect delay and crosstalk noise savings respectively for all the nets in the design. Columns six and eight indicate the savings of the net yielding maximum gain in terms of interconnect delay and crosstalk noise respectively. Comparing the game theoretic results from Tables 3.1. and 4.1., it can be noticed that the transmission line models developed in this chapter for interconnect delay have sufficient accuracy and improved run times, and hence can be used in optimization problems requiring extensive interconnect delay computations on net segments.

CHAPTER 5

GATE SIZING

In this chapter, we develop a framework for multi-metric optimization which is capable of optimizing various conflicting design parameters. We model the post-route gate sizing for simultaneous interconnect delay and crosstalk noise optimization as a game theoretic optimization problem and solve it using Nash equilibrium theory. The crosstalk noise induced on a net depends on the size of its driver gate and the size of the gates driving its coupled nets. Increasing the gate size of the driver increases the noise induced by the net on its coupled nets, whereas increasing the size of the drivers of coupled nets increases the noise induced on the net itself, resulting in a cyclic order dependency leading to a conflicting situation. Game theory inherently models the competition and is well suited for conflicting situations. The gates of the design are modeled as the players, the possible set of gate sizes for each gate is modeled as the strategy set, and the normalized geometric mean of interconnect delay and crosstalk noise is modeled as the payoff function of the normal form game. We have implemented two different strategies in which games are ordered according to (i) the noise criticality, and (ii) delay criticality of nets. The time and space complexities of the proposed gate sizing algorithm are linear in terms of the number of gates in the design. Also, we have provided a mathematical proof of existence for Nash equilibrium solution for the proposed gate sizing formulation.

5.1 Problem Definition

The problem of gate sizing can be defined as finding the optimal sizes for all gates in the circuit such that the overall interconnect effects (delay and crosstalk noise in this chapter) are minimized without need for rerouting or increase in area overhead. The coupling capacitance is responsible for the majority of the deep submicron effects. Hence, it is important to extract the coupling capacitance of nets with high accuracy. The coupling capacitance of a net depends the length of overlap and spacing between adjacent nets. This information can be efficiently extracted at post-routing phase. The coupling noise induced on a net depends on the size of its driver, driven and aggressor gates. Also, the interconnect delay is a function of the gate sizes, and the input and load capacitances. Equations 2.3 and 2.10 emphasize that the gate sizes directly control the interconnect delay and crosstalk noise in terms of the driver resistances, gate and coupling capacitances. Hence, calculating the optimal gate sizes can effectively reduce the crosstalk noise and interconnect delay in deep submicron designs. Gate sizing can be performed at post-route level by utilizing the existing fill-space. In our approach, we incrementally scale the gate sizes to utilize the available fill-space such that the routed resources in adjacent regions are not disturbed. Hence, our approach will neither result in area overhead nor need re-routing of the design.

5.2 Motivation for Gate Sizing Using Game Theory Model

The coupling noise induced on a net depends on the size of the victim and the aggressor gates. When the size of the victim gate is increased, the crosstalk noise on the victim net decreases, but increases the noise induced by it on the aggressor nets. Hence, the aggressor gates need to be sized-up in order to reduce the effect of sized-up victim driver. Increasing the size of aggressors will increase the noise induced on the victim net, resulting in a cyclic order dependency leading to a conflicting situation. It is pointed out in [10] that solving the post-route gate sizing problem for crosstalk noise optimization is difficult due to this conflicting nature of the problem. It is possible to develop a framework based on game theory which lends itself well to modeling such conflicting situations. In a game theoretic model involving convex payoff functions, it has been shown in [16] that the Nash equilibrium solution always exists and tends to yield globally optimal solutions [17].

As the size of a gate increases, the interconnect delay through the driven net decreases and the crosstalk noise induced on the adjacent nets increases (a convex function). Traditionally, this problem is modeled using crosstalk noise as the objective function, while maintaining interconnect delay as a constraint or vice versa. Through use of the game theoretic formulation and the Nash equilibrium function, it is possible to achieve simultaneous optimization of multiple metrics with conflicting objectives. Since, interconnect delay and crosstalk noise within a circuit are conflicting in nature, leading to a convex objective function, and a convex objective function is a requirement for the Nash equilibrium function to yield good results, there is a good motivation. The performance of the proposed algorithm is compared with that of simulated annealing and genetic search in order to illustrate the elegance of game theoretic solutions for problems with conflicting objectives. It is shown in Section 5.7 that the proposed approach yields better results than simulated annealing, genetic search, and Lagrangian relaxation under the assumptions of the same models, setup, parameters and the objective function.

5.3 Game Theoretic Gate Sizing for Multi-metric Optimization

In this section, we formulate and develop a methodology for simultaneous optimization of interconnect delay and crosstalk noise using gate sizing. The problem of simultaneously optimizing interconnect delay and crosstalk noise is being attempted for the first time. This problem is complex to solve because of the conflicting nature of the interconnect delay and crosstalk in any given circuit. Given a placed and routed design, we model a one-shot game for each interconnect net and solve it to size a particular number of gates associated with the game. The order in which the nets are chosen to create the one-shot games is critical in deciding the percentage optimization achieved in terms of interconnect delay and crosstalk noise. The interconnect nets can either be ordered according to its noise criticality or the delay criticality with respect to the other nets in the design. We have investigated these two types of orderings and developed a game theoretic framework as given below.

5.3.1 Approach 1: Gates Ordered Based on Noise Criticality

We develop an optimization task which can be performed after the place and route phase of the given gate-level netlist. The interconnect resistance, capacitance, inductance, and the set of aggressor drivers along with their coupling capacitance is extracted for each net from the SPEF netlist of the routed design. We extract the length of interconnect wires, the length of overlaps of each net with its set of aggressor nets and their wire spacings from the routed design exported in DEF format. The multi-terminal net is considered as different nets with same driver and different receivers. The gates of the design are ordered according to the noise criticality of the driven nets. Recent works on gate sizing for crosstalk noise optimization [10, 31], employ a crosstalk noise estimator in their noise optimization engines for identifying the noise critical nets of the given design. This is a time consuming process. In this work, rather than estimating the noise induced on each net, we rank the nets relatively, to indicate whether a net is more noise critical than an other net or vice versa. The coupling capacitance between any two nets is proportional to the length of their overlap and inversely proportional to the square of the distance of their separation [57]. Hence, for each net, we define a score as

$$\sum_{\forall aggressors} \frac{\text{(length of their overlap)}}{\text{(spacing)}^2}$$

The nets with high score values are ranked higher to indicate that they are more noise critical. The nets are sorted in a list according to their score values. The most critical net will form the head of the list while the least critical net will form its tail. The gates are considered for their size optimization in the order of the ranks of their driven nets. Referring to the Equation 2.3, the interconnect delay of a net depends on the size of its driver and receiver gates. The crosstalk noise induced on a net depends on the size of all aggressor gates and its victim driver. For any given net, there can be many potential aggressor gates. It is indicated in [10] that it is virtually not possible to consider the noise effects of all the aggressor gates on a given net. Hence, we consider the effects of two most affecting aggressor gates while sizing the gates related to the given net. However, the algorithm developed in this work is not limited to two aggressor gates and can be easily extended to consider the effects of more than two aggressor gates for each net. In addition, we show from experimental results that consideration of two most affecting aggressor gates is sufficient to take coupling effects into account.

A game is modeled for each net in the order of their sorted list. The game created is a 4-player game whose players are the driver, the receiver and the two most affecting aggressor gates of the chosen net. The two most affecting aggressor gates are chosen among its pool of aggressors based on the fraction of their contribution to the score of the given net. The two aggressor gates which contribute to the majority of the score value are selected. We have used normal form formulation to mathematically represent and solve the game. A normal form game consists of a set of N players labeled $1, 2, \ldots, N$, such that each player i has: (i) a choice set S_i called strategy set of player i; its elements are called strategies, and (ii) a payoff function $P_i : S_1 \times S_2 \times .... \times S_N \rightarrow \Re$, assigned to each strategy chosen by the player i with respect to other players. The strategy set and the payoff matrix of all the individual players are sufficient to solve the normal form game. All the players play simultaneously without any knowledge about other the players' actions. In other words, each player simultaneously chooses a strategy $s_i \in S_i$ such that the corresponding payoff is minimized with respect to the payoffs of the other players. The equilibrium solution of the game is computed using the Nash equilibrium theory.

The strategy set for each gate is modeled as the set of various possible gate sizes with which it can be scaled. The scalable gate sizes for each gate are chosen such that its replacement in the design does not result in re-routing. The maximum scalable gate size depends on two factors: (i) available free space surrounding the gate in the design, and (ii) drive strengths available for a gate type in the standard cell library. For a gate at specific location in the design, all the gate sizes supported by the standard cell library cannot be used as its strategies. Some of the gate sizes cannot be fitted within the available free space without disturbing the routed nets surrounding it. Hence, the number of strategies

Algorithm 5.1. Gate Sizing Algorithm for Interconnect Delay and Crosstalk Noise Optimization

```
Input: Placed and routed design
Output: Optimized gate sizes
  Algorithm:
  extract the net parasitics from SPEF file
  for all gates do
    determine aggressors();
      % extract aggressor gates from SPEF file
    mark the gate as un-played
  end for
  for all nets do
    extract the overlapping lengths and spacing between the adjacent nets from DEF file
    calculate scores();
    sort scores();
  end for
  while there exists an unsized gate do
    select an un-played noise critical net $i$ from the sorted list
    identify two main aggressor gates for net $i$ in the order of their coupling effects
    create a 4-player game with driver, receiver, and two main aggressor gates of net $i$
    for gate $g_k$ among the four players do
      if $g_k$ is marked as sized then
        strategy set of $g_k$ ← calculated Nash size;
      else
        strategy set of $g_k$ ← determine strategies();
      end if
    end for
    cost-matrix ← payoff(four players, strategies)
      % for payoff function, see Algorithm 5.2.
    optimized-size ← nash-solution(four players, payoffs)
      % for Nash equilibrium solution, see Algorithm 5.3.
    mark the four played gates as sized
    mark the net $i$ as played
  end while
  return: optimized Nash sizes of gates
```

available for each gate is always less than the number of drive strengths available for that gate type in the standard library. The minimum size of the gates is set to minimum drive strength available in the standard cell library. If a gate involved in the current game is marked as sized due its participation in earlier played games, its strategy set is modeled as a singleton set consisting of its calculated Nash width. The strategy set for other players involved in the game is modeled as the available gate sizes in the standard cell library between its minimum and maximum gate sizes.

The payoff function tries to capture the interaction between the four gates identified as the players of the game. For the chosen net, its delay D and the maximum crosstalk noise N are calculated using the Equations 2.3, and 2.7 respectively. These values are calculated for all strategies of a gate considering the strategies of other players of the game. The delay and crosstalk noise values obtained for each strategy of a player are normalized with respect to a particular strategy. The normalization is performed to transform the delay and crosstalk noise values into dimensionless quantities so that they can be easily correlated with each other. The payoff function is modeled as the geometric mean of normalized delay and noise values of the players so as to give equal weight to both interconnect delay and crosstalk noise.

Algorithm 5.2. Algorithm for Payoff Matrix Calculation

```
Input: Number of Players $N$, Strategy set $S$
Output: Payoff matrix
  for all players $i \in 1$ to $N$ do
    for all strategy combinations $S^j = \{s_1^j, \ldots, s_N^j\}$, where $(s_1^j \in S_1), \ldots, (s_N^j \in S_N)$ do
      calculate the delay using Equation 2.3
      normalize the delay w.r.t first strategy combination
      calculate the crosstalk noise using Equation 2.7
      normalize the noise w.r.t first strategy combination
      $P[i, S^j] \leftarrow$ Geometric mean of normalized noise and delay
    end for
  end for
  return: payoff matrix $P$ $\forall$ strategy combinations
```

The Nash equilibrium is evaluated for the chosen net and the game is played out. The four gates that participated in the played game are flagged as “sized” and their sizes are set equal to the calculated Nash sizes. The chosen net, for which the game is played out, is tagged as played net and is removed from the sorted list of ordered nets. A new net located at the head of the sorted list is selected to play the next game. The four players corresponding to the selected net are identified. The strategy set of the gates which are marked as “sized” are defined as a singleton consisting of only its calculated Nash width.

The strategy sets for other players of the game are identified as described above and the Nash equilibrium of the game is evaluated. This process of creating and playing the sequential games is repeated until all the gates of the design are marked as sized. It can be noted that the number of games played is always less than the total number of gates in the design. The pseudo-code of the complete gate sizing algorithm developed in this work is shown in Algorithm 5.1..

Algorithm 5.3. Algorithm for Nash Equilibrium Solution

```
Input: Number of players $N$, Payoff matrix $P$
Output: Nash solution
  for all players $i$ do
    for all payoffs of player $i$ do
      find $s_i^*$ such that,
      $P_i(s_1^*, \dots, s_i^*, \dots, s_N^*) \geq P_i(s_1^*, \dots, s_i, \dots, s_N^*)$
      $s_i^*$ is the Nash strategy for player $i$
    end for
  end for
  Nash-solution $S^* = \{s_1^*, \dots, s_N^*\}$
     % set of optimized strategies for all $N$ players
  return: Nash solution $S^*$
```

5.3.2 Approach 2: Gates Ordered Based on Delay Criticality

The ordering of nets in the sorted list dictates the order in which the gates are considered for their size optimization. In section 5.3.1, the interconnect wires are sorted in a list based on the noise criticality of the nets. Hence, the approach outlined in section 5.3.1, yields slightly better optimization of crosstalk noise than interconnect delay, while simultaneously optimizing both delay and noise. In this section, we investigate a strategy wherein delay is considered as higher criticality than noise, while simultaneously optimizing delay and noise. It is interesting to note that both methods yield significantly better optimization of both delay and noise compared to other methods. The designer can choose either of the strategies based on the need. The difference between the two strategies is the way in which the sorted list is created. After the design is placed and routed, the path delays of all the paths in the design are estimated, and are sorted into a list based on their delay criticality.

The most delay critical path is chosen to create games for gate size optimization. The games are created for each net in the chosen path in the order from its primary output to primary inputs. As an example for illustration, consider the chosen path to consist of four gates: A, B, C and D, in successive transition connected with nets: $1, 2$ and 3, respectively. The gate A is driven by primary inputs and gate D drives a primary output. In order to consider the down-stream load capacitance, the net 3 connecting gates C and D should be optimized before the nets 1 and 2. Thus, the games are played in the order of net 3 followed by net 2 followed by net 1. The game formulated for each net involves its driver, receiver and its two most affecting aggressor gates as its players. The two most affecting aggressor gates for the net and its strategies are identified as indicated in section 5.3.1. After the games are played for all the nets of the chosen critical path in its direction of primary output to primary inputs, the next critical path in the sorted list is selected to play games. This process of creating games is repeated until the sorted list is empty.

5.4 Time and Space Complexity of Proposed Gate Sizing Algorithms

The worst case time complexity of evaluating Nash equilibrium for a general M-player game with S strategies for each player is given as $O(M * S^M)$ [52]. Referring to Section 5.3.1, we have modeled the problem of gate sizing for simultaneous interconnect delay and crosstalk noise reduction as a game with four players. For each gate type, the number of different drive strengths available for it in the standard cell library act as its maximum number of strategies. We have used a standard cell library containing gates with four different drive strengths, built on TSMC 180nm design rules. Even though, there are four different drive strengths available in the library, the scalable sizes for each gate depends on the location and the free space surrounding it in the design. Hence, the number of strategies available for each gate is less than or equal to four. Thus, the complexity of calculating the Nash equilibrium for a single game played is given by $O(4 * 4^4)$. The games are played repeatedly until all the gates are sized. In any game, if a gate is marked as sized, then its strategy set is modeled to have single strategy consisting of its calculated Nash size. When a game is played, the outcome is finding the best gate sizes for all the players which are not sized before. The number of games played is less than the total number of gates in the design because after each game is played out, at least one new gate will be sized. Hence, the overall time complexity of all the games played can be mathematically given as

$$O\left(N_{gates} * 4 * 4^4\right) \approx O\left(N_{gates}\right)$$

where N_{gates} is the total number of gates in the given design. It can be noticed that the time complexity of the proposed algorithm is *linear* and is proportional to the total number of gates in the design.

The space complexity of the proposed algorithm is dependent entirely on the number of gates in the design and the space complexity of the payoff matrix. The space complexity of a payoff matrix depends on the number of strategies for each player playing the game. As the games are played sequentially, the total space required by all the games is equal to the space complexity of a game involving players with the maximum number of strategies. Mathematically, the space complexity required by all the payoff matrices is given as $O(S_1 * S_2 * S_3 * S_4) \leq O(4 * 4 * 4 * 4)$, where $S_1, S_2 S_3$, and S_4 are strategy sets of 4-player game involving the players with maximum strategies. Hence, the space complexity of the proposed algorithm is given as

$$O\left(N_{gates} + 4 * 4 * 4 * 4\right) \approx O\left(N_{gates}\right)$$

5.5 Proof of Existence of Nash Equilibrium for the Proposed Gate Sizing Formulation

In this section, we provide the proof of existence of Nash equilibrium in the case of gate sizing problem for simultaneous optimization of interconnect delay and crosstalk noise. As the gate size of a gate increases, the interconnect delay decreases and the coupling capacitance increases resulting in a convex payoff function. Let $G = \{S_1, \ldots, S_n; f_1, \ldots, f_n\}$

be a game with each player $i \in N$ having a strategy set S_i containing its possible gate sizes and its payoff given by f_i. We have modeled the strategy set S_i for each player as a non-empty, compact set of a finite dimensional Euclidean space. Because of the convex nature of the interconnect delay and crosstalk noise, the modeled payoff function f_i becomes upper semicontinuous on $S = \Pi_{i=1}^{N} S_i$ and for any fixed $u_i \in S_i$, the function $f_i(u_i, .)$ is a lower semicontinuous on $S_{(-i)}$ [16]. For any $u \in S$, the best reply or the expected payoff $B_i(u)$ is also convex. According to Kakutani's fixed point theorem [61], the game G has at least one Nash equilibrium point if the graph

$$G_B = \{(x, y) : x \in S, y \in B(x)\}$$

is closed.

Lets assume that it is not closed. Then, $\exists (x^0, y^0) \notin G_B$, such that every neighborhood (in $S \times S$) of (x^0, y^0) contains a point of G_B.

$\because x^0$ is a gate size, it has to be one of those from the set of possible gate sizes for the given player in order to satisfy the DRC rules of the used process technology.

$\therefore x^0 \in S \Rightarrow y^0 \notin B(x^0)$

In other words, for at least one gate playing the game (say gate 1), there is an $y_1^1 \in S_1$ such that

$$f_1(y_1^1, x_2^0, \ldots x_n^0) \geq f_1(y_1^0, x_2^0, \ldots x_n^0) \tag{5.1}$$

Let F be a function such that $F : S^2 \rightarrow \Re$ and given as

$$F(x, y) = f_1(y_1^1, x_2, \ldots x_n) - f_1(y_1, x_2, \ldots x_n)$$

Since f_i is upper semicontinuous on S and $f_i(u_i, .)$ is lower semicontinuous on S_{-i}, F is lower semicontinuous and $C = \{(x, y) \in S^2 : F(x, y) \leq 0\}$ is closed. Hence, for any $(\bar{x}, \bar{y}) \in G_B, F(\bar{x}, \bar{y}) \leq 0$. But, by Equation 5.1, $F(x^0, y^0) \geq 0$, contradicting the closedness of C. Thus, there is a point $s^* \in S$ such that $s^* \in B(s^*)$, which is a Nash equilibrium point.

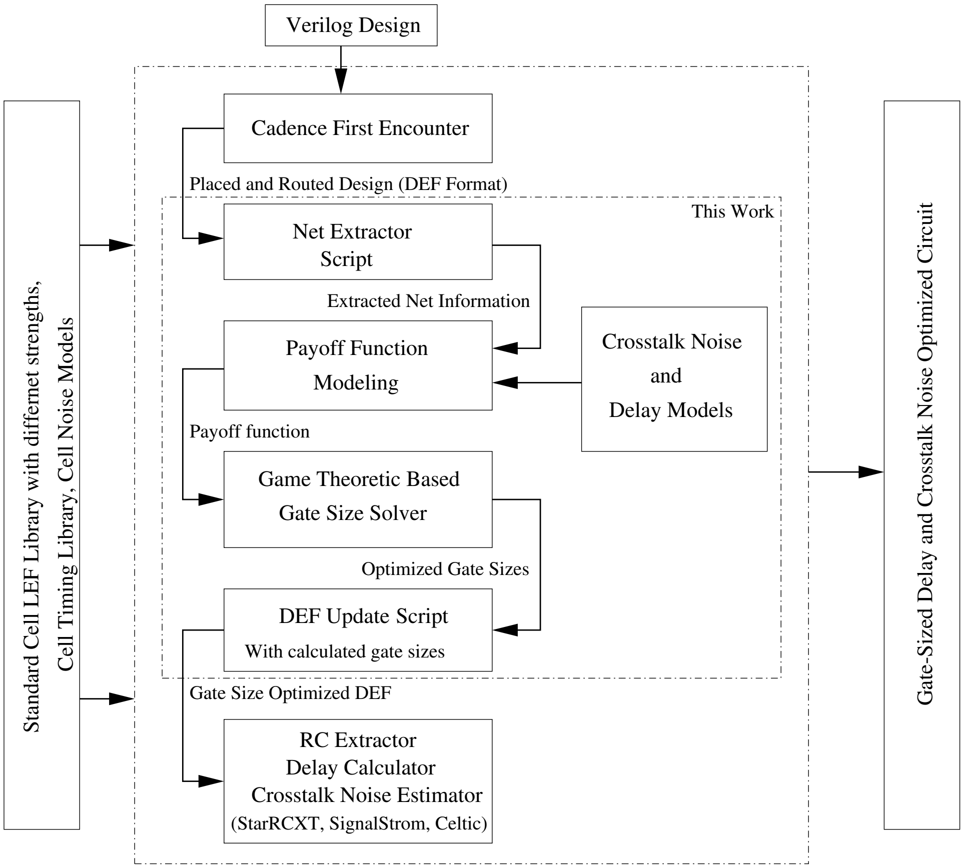

Verilog Design
Standard Cell LEF Library with differnet strengths, Cell Timing Library, Cell Noise Models
Cadence First Encounter
Placed and Routed Design (DEF Format)
This Work
Net Extractor Script
Extracted Net Information
Payoff Function Modeling
Crosstalk Noise and Delay Models
Payoff function
Game Theoretic Based Gate Size Solver
Optimized Gate Sizes
DEF Update Script
With calculated gate sizes
Gate Size Optimized DEF
RC Extractor
Delay Calculator
Crosstalk Noise Estimator
(StarRCXT, SignalStrom, Celtic)
Gate-Sized Delay and Crosstalk Noise Optimized Circuit

Figure 5.1. Integration of Proposed Gate Sizing Algorithm in the Design Flow

5.6 Design Flow

The design flow for obtaining an optimally gate sized circuit from a verilog/VHDL description is shown in Figure 5.1.. The behavioral verilog/VHDL description is synthesized on to a library of standard cells and given as input to the design flow. The standard cells are placed and routed in accordance with the synthesized code using any standard cell place and route tool. We have used the First Encounter™ RTL-to-GDSII tool from Cadence® Design Systems to perform the placement and routing of gate-level design. The parasitics from the routed design are exported in SPEF format with the help of StarRCXT from Synopsys Inc. A lex and yacc script is developed to read the SPEF netlist and extract the values of interconnect resistance, capacitance, inductance and coupling capacitances along with their set of aggressor gates for all the nets of the design. Also, a gawk script is written to extract the information about the length of overlaps between two given nets along with their distance of separation from the DEF netlist. The models used for Interconnect delay and crosstalk noise, and the modeling of the payoff function are described in Sections 2.5 and 5.3.1 respectively. The payoff function is used by the game theoretic based gate size solver described in Algorithm 5.1. to minimize the interconnect delay and crosstalk noise of individual nets in the order of their noise criticality. The optimized gate sizes resulted from our solver are used to update the routed design. We have developed another gawk script which scales the gates in the original DEF routed design according to their calculated optimized gate sizes. It should be noted here that the resulting optimized design does not require re-routing since the possible gate sizes drafted for each gate are within its limits to satisfy the design rules of the used process technology.

5.7 Experimental Results

The game theoretic gate size solver described in Algorithm 5.1. was implemented in C and executed on a UltraSPARC-IIe 650MHz, 512MB Sun Blade 150 system running Solaris 2.8. The ASIC designs on which we tested our algorithm were obtained from Opencores [62]. A standard cell library containing 10 logic cells with 4 different drive strengths based on a 6-Metal layer, 180nm technology has been developed and used. ASIC designs, written in behavioral VHDL/Verilog are converted to structural VHDL/Verilog using the standard cells in the library with the help of BuildGates, an RTL synthesis tool of Cadence design systems. We have modified the ASIC designs such that all the blocks in the design are flattened to standard cells in the library without maintaining the hierarchy. The on-chip memory modules are realized as register arrays with D-flipflop as basic building units. The structural VHDL/Verilog design is used as input by Cadence First Encounter to develop the floorplan. We have set the option of row utilization to 70% for all the designs so as to allow some gate size scaling. The design is then placed and routed using Amoebaplace and Nanoroute respectively, which are part of Cadence First Encounter tool. The final placed and routed design is then exported in DEF format.

The parasitic information from the routed design is extracted in SPEF format using Synopsys StarRCXT. The interconnect resistance, interconnect capacitance, interconnect inductance, coupling capacitances along with their aggressor drivers is extracted from the SPEF file and is given as input to the game theoretic gate size solver. The length of overlap with the aggressor nets and their spacing is extracted from the DEF file and is also given as input. The calculated gate size for each gate is used to update the original DEF file to generate an optimized DEF file. It can be noted that the optimized DEF file is created with the help of a gawk script and verified for DRC rules. The design is not rerouted to generate the optimized DEF file. The interconnect delay and crosstalk noise are estimated using Cadence Signalstorm and CelticIC tools respectively with their robust models, and not using the analytical models used in the dissertation.

The works reported in literature solve the problem of gate sizing for crosstalk noise optimization under delay constraints. In this work, we have solved the problem of gate sizing for simultaneous optimization of crosstalk noise and interconnect delay. Hence, in order to compare our results, we have implemented simulated annealing and genetic search for simultaneous optimization of crosstalk noise and interconnect delay, and executed it on same Solaris machine with same set of inputs and parameters. The simulated annealing algorithm was implemented and we experimented to obtain the best results in terms of optimization of interconnect delay and crosstalk noise. The set of possible gate sizes is calculated as indicated in Section 5.3.1. In each move of the simulated annealing process, a gate is randomly selected and its size is randomly assigned from the set of its possible gate sizes. The cost function is defined as the geometric mean of interconnect delay and crosstalk noise summed over all the nets. The initial temperature is determined by finding the average change in the cost for a set of random moves from the starting configuration and selecting the temperature which leads to an accept probability of 0.95. The number of moves per temperature for each design is set to 20 times the total number of gates in the design. This is done so as to allow at least 10 to 15 moves on the average for each gate before settling for its solution. The up-hill moves are accepted with a probability of $e^{-\frac{\delta C}{T}}$, where δC is the change in the cost and T is the current temperature of the iteration. The temperature is cooled at the rate of 0.95.

The gate sizing problem for simultaneous optimization of interconnect delay and crosstalk noise is modeled as a genetic search mechanism and solved using GALib [64]. The initial population contains the gates represented by their corresponding gate sizes as used in the original unsized design. Each individual in the population, called a chromosome, is represented as a set of two integers indicating the gate number and the gate size assigned to it. The chromosomes evolve through successive iterations called generations. During each generation, the chromosomes are evaluated for their fitness test. We have defined the fitness criterion as the deviation of the crosstalk noise and interconnect delay of each gate from their worst-case values. The chromosomes with lower values of crosstalk noise and interconnect delay are given higher fitness values. We have used the steady-state genetic algorithm available as a part of GALib library to generate overlapping populations which retains its 30% of fittest chromosomes in its new generations. The mutation process for a chromosome is defined as randomly selecting a gate size from its set of possible gate sizes. The new chromosomes are created using single point crossover and are validated against their set of possible gate sizes. The selection process of a chromosome is adopted by the roulette wheel selection approach. We have set the convergence-of-population as the stopping measure for the evolution of generations.

Experimental results are provided in Tables 5.1. and 5.2.. Table 5.1. indicate the average interconnect delay and crosstalk noise savings of all nets of the design, whereas the Table 5.2. indicates the critical path savings in terms of interconnect delay and crosstalk noise. Referring to Table 5.1., the first column indicates the name of the open core design and the second column indicates the corresponding number of gates in the design. Columns three, six, nine, and twelve indicate the running times of genetic search, simulated annealing, game theoretic approach based on noise criticality (GT-NC approach) and game theoretic approach based on delay criticality (GT-DC approach) respectively.

Table 5.1. Average Savings for Simultaneous Optimization of Interconnect Delay and Crosstalk Noise During Gate Sizing*

Open Core Design [62]	Total Gates	GS Approach[1]			SA Approach[2]			GT-NC Approach[3]			GT-DC Approach[4]		
		Run Time (mins)	% Average Savings		Run Time (mins)	% Average Savings		Run Time (mins)	% Average Savings		Run Time (mins)	% Average Savings	
			Delay	Noise		Delay	Noise		Delay	Noise		Delay	Noise
Mult	428	28.40	3.06	3.21	10.28	4.83	3.74	1.13	4.18	5.34	1.06	6.32	3.98
PCI	7882	158.41	5.29	4.64	38.83	8.14	10.68	4.29	10.39	16.49	4.16	12.11	13.89
ATA	21781	352.92	5.78	7.30	64.15	14.12	16.81	10.49	20.31	23.91	10.67	21.83	20.79
RISC	34172	587.12	4.72	7.91	79.61	12.92	13.95	13.31	15.28	14.05	13.46	16.72	13.48
AVRμP	41274	716.72	8.31	8.92	112.48	15.67	16.78	15.79	21.41	22.96	15.13	24.41	18.63
P16C55	52128	1089.52	5.94	6.11	159.76	13.35	17.23	19.98	16.96	21.91	20.13	19.40	20.07
T80μC	69973	1426.32	6.43	7.86	220.57	17.42	21.63	27.67	19.86	25.24	27.14	20.74	24.83
Average			5.65	6.56		12.35	14.40		15.48	18.56		17.36	16.52

* No area overhead for all four approaches. The percentage values indicated are w.r.t placed and routed design without gate sizing.

[1] GS Approach: Genetic search based gate sizing for simultaneous optimization of interconnect delay and crosstalk noise

[2] SA Approach: Simulated annealing based gate sizing for simultaneous optimization of interconnect delay and crosstalk noise

[3] GT-NC Approach: Game theoretic gate sizing with gates ordered based on noise criticality for simultaneous optimization of interconnect delay and crosstalk noise

[4] GT-DC Approach: Game theoretic gate sizing with gates ordered based on delay criticality for simultaneous optimization of interconnect delay and crosstalk noise

Table Legend: *Average Savings*: Average savings of all the nets in the entire design; *Runtime*: running time of each algorithm.

Table 5.2. Critical Path Savings for Simultaneous Optimization of Interconnect Delay and Crosstalk Noise During Gate Sizing *

Open Core Design [62]	Total Gates	GS Approach[1]			SA Approach[2]			GT-NC Approach[3]			GT-DC Approach[4]		
		Run Time (mins)	% Crit. Path Savings		Run Time (mins)	% Crit. Path Savings		Run Time (mins)	% Crit. Path Savings		Run Time (mins)	% Crit. Path Savings	
			Delay	Noise		Delay	Noise		Delay	Noise		Delay	Noise
Mult	428	28.40	4.98	4.16	10.28	5.71	6.68	1.13	5.59	7.86	1.06	8.41	7.01
PCI	7882	158.41	7.84	6.02	38.83	14.29	17.51	4.29	17.11	25.29	4.16	20.30	24.12
ATA	21781	352.92	9.47	6.51	64.15	15.91	18.43	10.49	24.78	27.12	10.67	25.99	24.39
RISC	34172	587.12	4.39	8.10	79.61	19.41	17.50	13.31	21.16	19.96	13.46	23.09	17.71
AVRμP	41274	716.72	7.49	9.91	112.48	20.69	22.54	15.79	27.19	28.95	15.13	35.52	22.31
P16C55	52128	1089.52	7.03	5.67	159.76	16.75	17.49	19.98	21.54	23.16	20.13	25.11	21.69
T80μC	69973	1426.32	3.95	10.64	220.57	20.78	25.41	27.67	24.21	28.92	27.14	26.81	26.22
Average			6.45	7.29		16.22	17.94		20.23	23.04		23.60	20.49

* For table footnotes, please refer to footnotes given under Table 5.1.. *Crit. Path Savings*: Savings on the critical net of the design

The columns four and five of Table 5.1. indicate the average interconnect delay and crosstalk noise savings for all the nets of the design obtained by genetic search approach. Columns seven and eight represent these values for simulated annealing approach, while columns ten and eleven indicate for GT-NC approach and columns thirteen and fourteen indicate for GT-DC approach respectively. The columns of Table 5.2. represent the same corresponding values obtained for critical path of the respective design. The experiments were conducted such that the area overhead is zero in all four approaches. Referring to Table 5.1., genetic search shows 5.65% and 6.56% of average interconnect delay and crosstalk noise improvements. Simulated annealing shows 12.35% and 14.40% of average interconnect delay crosstalk noise improvements. In comparison, GT-NC approach shows 15.48% and 18.56% of average interconnect delay and crosstalk noise improvements, while GT-DC approach shows improvements of 17.36% and 16.52% respectively. Referring to Table 5.2., genetic search results in 6.45% and 7.29% and simulated annealing results in 16.22% and 17.94% in terms of critical path interconnect delay and crosstalk noise savings respectively. In comparison, GT-NC approach yields 20.23% and 23.04% improvements on critical nets in terms of interconnect delay and crosstalk noise, while GT-DC approach yields 23.60% and 20.49% improvements respectively. Also, we have observed that interconnect power consumption follows the same trend as that of crosstalk noise. The decrease in coupling capacitance results in a smaller switched capacitance and thereby would result in lesser power dissipation. The game theoretic gate size solver, in addition to outperforming genetic search and simulating annealing in terms of interconnect delay and crosstalk noise savings, has significantly smaller run times. Hence, our approach is scalable and favorable to handle the complexity of large SOC designs.

To enable a direct comparison of our work with the recent work reported in [31], we have modified our game theoretic approach 1, given in Section 5.3.1 to minimize the crosstalk noise under delay constraints. The work developed in [31] is a Lagrangian relaxation based gate sizing approach for reducing the crosstalk noise under delay constraints. The developed algorithm is iterative and makes use of a coupling graph developed based on the coupling

Table 5.3. Crosstalk Noise Optimization Under Delay Constraints During Gate Sizing *

Open Core Design [62]	Total Gates	Number of Noise Violations Noise Threshold = $0.15V_{dd}$	
		Lagrangian Based [31]	Game Based [This Work]
Mult	428	7	5
PCI	7882	23	11
ATA	21781	97	26
RISC	34172	148	53
AVRμP	41274	181	59
P16C55 μC	52128	239	67
T80μC	69973	289	83

* Here, we have used the number of noise violations as the metric, since it is used in [31] as their algorithm evaluation criteria.

capacitances. For both game theoretic and Lagrangian relaxation [31] approaches, we have used the delay values obtained from simultaneous optimization of interconnect delay and crosstalk noise as the set of delay constraints, so as to ensure a tighter constraint set for both approaches. The Lagrangian relaxation problem for post-layout crosstalk noise reduction is formulated as developed in [31] and solved using LANCELOT [66]. The game theoretic approach 1, given in Section 5.3.1, is modified such that the strategies which do not satisfy the delay constraints are pruned out from their respective strategy sets. The payoff function is then modified to account solely for crosstalk noise induced on the net under consideration. Table 5.3. shows the comparison of game theoretic and Lagrangian relaxation based approaches indicated in terms of number of noise violations for each design. Noise violations are expressed as the number of nets which have an induced noise exceeding a threshold noise set to $0.15V_{dd}$. It can seen that our approach results in significantly fewer faults when compared to the Lagrangian relaxation.

5.8 Conclusions

Game theory allows the simultaneous optimization of multiple metrics in the context of conflicting objectives leading to a convex objective function in the problem formulation. This essentially makes it possible to use game theory for simultaneous optimization of interconnect delay and crosstalk noise. Optimizing both interconnect delay and crosstalk noise is extremely critical in deep submicron and nano regime circuits. The proposed method results in a linear time algorithm with significantly better results than genetic search, simulated annealing and Lagrangian relaxation, making this work an important contribution.

CHAPTER 6

INTEGRATED GATE AND WIRE SIZING

In this chapter, we develop a new post-layout integrated gate and wire sizing algorithm for simultaneous optimization of interconnect delay and crosstalk noise. The problem of post-layout gate and wire sizing is modeled as a normal form game and solved using Nash equilibrium. The crosstalk noise induced on a net depends on its wire size, its driver size and the sizes of gates driving its coupled nets. It is reported in [10] that solving the problem of crosstalk noise optimization at post-route level is difficult due to the cyclic dependency, resulting in a conflicting situation. Game theory provides a natural framework for handling conflicting objectives and allows simultaneous optimization of multiple parameters. The formulation of a convex objective function is a requirement in order to obtain better optimization in a game theoretic framework. This property is exploited to solve the cyclic dependency of crosstalk noise on its gate and wire sizes, while modeling the problem of simultaneous optimization of interconnect delay and crosstalk noise, which again are conflicting in nature. A game is modeled with the crosstalk noise and interconnect delay of the chosen net as the players, the possible gate and wire sizes as the strategy set and the analytical expressions for crosstalk noise and interconnect delay as their respective expected payoffs. We have implemented two different strategies in which the games are ordered according to (i) the noise criticality, and (ii) the delay criticality of nets. The time and space complexity of the proposed integrated sizing algorithm is linear in terms of the number of gates and wires in the design.

In Chapters 3 and 5, we have independently solved the problem of simultaneous optimization of interconnect delay and crosstalk noise using wire sizing and gate sizing respectively. In Chapter 3, we have modeled the problem of post layout wire sizing as a number of 3-player games with wire segments as the players, possible wire sizes as the strategy set and normalized geometric mean of interconnect delay and crosstalk noise as the payoff function. In Chapter 5, we have modeled the problem of post layout gate sizing as a number of 4-player games with gates as the player, possible gate sizes as the strategy set and normalized geometric mean of interconnect delay and crosstalk noise as the payoff function. On contrary, in this chapter, we address the problem of integrated gate and wire sizing for simultaneous optimization of interconnect delay and crosstalk noise. The modeling of games for the integrated problem is completely different from that given in Chapters 3 and 5. Here, we have modeled the interconnect delay and crosstalk noise as the players of the game rather than the gates or the wire segments. Hence, every game created is a 2-player non-zero sum game. This modeling helps in improving the run time significantly.

6.1 Problem Definition

The problem of post-layout gate and wire sizing can be defined as follows: find the optimal gate and wire sizes such that the interconnect effects (interconnect delay and crosstalk noise in this work) are minimized under the given area constraints and without the need for rerouting any of the nets in the design. The parasitic resistance and capacitance of interconnect wires are highly dependent on the wire widths and gate sizes. The coupling capacitance is responsible for the majority of the deep submicron effects. The coupling noise induced on a net depends on its wire size, and the gate sizes of the driver, receiver and aggressor gates. The interconnect delay is also a function of gate and wire sizes (see Equation 2.3). Hence, calculating the optimal gate and wire sizes can effectively reduce both crosstalk noise and interconnect delay in deep submicron designs. Gate and wire sizing can be performed at post-route level by utilizing the existing fill-space. In this work, we incrementally scale the sizes of gates and wires to utilize the available fill-space, such that the routed resources in adjacent regions are not disturbed. Hence, our approach will neither result in area overhead nor need rerouting of the design.

6.2 Motivation for Integrated Gate and Wire Sizing

The coupling noise induced on a net depends on the wire size of the net and the sizes of the victim and aggressor gates. When the size of the victim gate is increased, the crosstalk noise on the victim net decreases, but increases the noise induced by it on the aggressor nets. Hence, the aggressor gates needs to be sized-up to reduce the effects of sized-up victim driver. Increasing the size of aggressors will increase the noise induced on the victim net, hence, resulting in a cyclic order dependency leading to a conflicting situation. Similarly, changing the wire size of a net will affect the wire sizes of the neighboring nets, resulting in conflicting behavior even in the case of wire sizing. It is reported in [10] that solving the crosstalk noise optimization problem at post-route level is difficult due to this conflicting nature of the problem. Game theory provides a natural framework for handling such conflicting situations. As the size of a gate and wire increases, the interconnect delay through the driven net decreases and the crosstalk noise induced on the adjacent nets increases (convex payoff function). In a game involving convex payoff functions, the game theory works better [16] and Nash equilibrium solution always exists and tends to achieve global optimal solutions [17]. It has been shown in [17] that the complexity of determining the Nash equilibrium lies between P and NP depending on the problem formulation. It is shown in Section 6.7 that the proposed approach yield better results than simulated annealing and Lagrangian relaxation under the assumptions of the same models, setup, parameters and the objective function.

6.3 A New Approach to Integrated Gate and Wire Sizing

In this section, we formulate and develop a methodology for simultaneous optimization of interconnect delay and crosstalk noise using gate and wire sizing. The problem of simultaneous optimization of interconnect delay and crosstalk noise can be solved independently using gate sizing followed by wire sizing or wire sizing followed by gate sizing with the help of methodologies developed in Chapters 5 and 3. However, it would be more advantageous to solve the gate and wire sizing problems in an integrated framework rather than in a sequential framework. In sequential framework, since the algorithms for gate and wire sizing are applied in sequence, one after the other, the full optimization potential of the individual algorithms cannot be achieved. The algorithm applied first would have more resources available for it to optimize than the following algorithms and hence, would result in sub-optimal solutions for algorithms applied following the first. Given a placed and routed design, we model a 2-player one-shot game for each interconnect net and solve it to size a particular number of gates associated with the game. The order in which the nets are chosen to create the one-shot games is critical in deciding the percentage optimization achieved in terms of interconnect delay and crosstalk noise. The interconnect nets can either be ordered according to its noise criticality or the delay criticality with respect to the other nets in the design. We have investigated these two type of orderings and developed a game theoretic framework as given below.

6.3.1 Approach 1: Gates Ordered Based on Noise Criticality

After the place and route phase of the design, the interconnect resistance, capacitance, inductance, and the set of aggressor drivers are extracted for each net from the Standard Parasitic Exchange Format (SPEF) netlist. We extract the interconnect wire lengths, wire widths, the length of overlaps of each net with its set of aggressor nets and their wire spacings from the routed design exported in DEF format. These values are used for calculating the coupling capacitances between the given nets and their aggressor nets. A multi-terminal net is considered as different nets with same driver and different receivers. Recent works on crosstalk noise optimization [10, 31] use a noise estimator in their optimization engines, for identifying the noise critical nets of the design. This is a time consuming process. The coupling capacitance between any two nets is proportional to the length of their overlap and inversely proportional to the square of the distance of their separation [57]. Hence, rather than estimating the noise induced on each net, we rank the nets relatively, to indicate whether a net is more noise critical than an other net or vice versa. We define a score for each net as given by

$$\sum_{\forall aggressors} \frac{\text{(length of their overlap)}}{\text{(spacing)}^2}$$

The nets with high score values are ranked higher to indicate that they are more noise critical. The nets are sorted in a list according to their score values. The gates and wires related to the net are considered for their size optimization depending on the position of the net in the sorted list, starting from the head. Referring to the Equation 2.3, the interconnect delay of a net depends on the wire size of the net, and the gate sizes of the driver and the receiver. The crosstalk noise induced on a net depends on the wire sizes of the net and its aggressor nets, and the gate sizes of the aggressors and the victim driver. For any given net, there can be potentially many aggressor nets. It is indicated in [10] that it is virtually not possible to consider the noise effects of all the aggressors on the given net. Also, the crosstalk noise model based on transmission lines used in this work can handle only two aggressors. Hence, we consider the effects of two most affecting aggressors while sizing the gates and wires related to the given net. However, the algorithm developed in this work is not limited to two aggressor nets and gates, and can be easily extended to consider the effects of more than two aggressors for each net. In addition, we show from experimental results that consideration of two most affecting aggressor nets and gates is sufficient to take coupling effects into account. Here, we would like the readers to note that the measured values of crosstalk noise and interconnect delay in experimental results are based on the commercial models of cadence tools and not using the analytical models used in this work. Hence, the savings reported in section 6.7 are not based on the coupling effects due to the consideration of two aggressor nets.

A 2-player nonzero-sum game is modeled for each net, i, in the order of their sorted list. The interconnect delay and crosstalk noise of the net i act as the two players of the game. We have used normal form formulation to mathematically represent and solve the game. A normal form game consists of a set of N players labeled $1, 2, \ldots, N$, such that

Algorithm 6.1. An Integrated Gate and Wire sizing Algorithm for Interconnect Delay and Crosstalk Noise Optimization

```
Input: Placed and routed design
Output: Optimized gate and wire sizes
  Algorithm:
  - Extract the net parasitics from SPEF file
  - Extract the overlapping lengths and spacing between the adjacent nets from DEF file
  - Mark all the nets as “unsized”
  for all Gates do
    Determine aggressors();
    - Mark the gate as “unsized”
  end for
  Calculate scores();
     % Calculates score for all nets of the design
  Sort scores();
     % Sorts all the net into a list according to their scores
  while There exists an “un-played” net in the list do
    - Select the most noise critical net $i$ from the sorted list
    - Create a 2-player game with interconnect delay and crosstalk noise as players.
    - Identify the two main aggressor nets, $a_1$ and $a_2$, in the order of their coupling effects
    on net $i$
    - Tag the nets $i, a_1$ and $a_2$ as the strategy determiners (a total of three wires for wire
    sizing)
    - Tag the driver and receiver gates of nets $i, a_1$ and $a_2$ as the strategy determiners (a
    total of six gates for gate sizing)
    strategy set $\leftarrow \emptyset$;
    for all $SD_k \in$ strategy determiners do
      if $SD_k$ is marked as “sized” then
        strategy set $\leftarrow$ strategy set $\bigcup$ calculated Nash size of $SD_k$;
      else
        strategy set $\leftarrow$ strategy set $\bigcup$ set of possible sizes of $SD_k$;
      end if
    end for
    cost-matrix $\leftarrow$ payoff(two players, strategy set);
       % for payoff function, see Algorithm 6.2.
    optimized-size $\leftarrow$ nash-solution(two players, payoffs);
       % for Nash equilibrium solution, see Algorithm 6.3.
    - Mark the six gates involved in the game as “sized”
    - Mark the nets $i, a_1$ and $a_2$ as “sized” and remove it from the sorted list
  end while
  return: Optimized Nash sizes for gates and net segments
```

each player p has: (i) a choice set S_p called strategy set of player p; its elements are called strategies, and (ii) a payoff function $P_p : S_1 \times S_2 \times .... \times S_N \rightarrow \Re$, assigned to each strategy chosen by the player p with respect to other players. The strategy set and the payoff matrix of all the individual players are sufficient to solve the normal form game. All the players play simultaneously without any knowledge about other players' play. In other words, the players simultaneously chooses a strategy $s_p \in S_p$ such that their respective payoff is maximized with respect to the payoffs of the other players. The equilibrium of the game is computed by using the Nash equilibrium theory.

Algorithm 6.2. Algorithm for Payoff Matrix Calculation

```
Input: Number of Players $N$, Strategy set $S$
Output: Payoff matrix
  for all players $i \in 1$ to $N$ do
    for all strategy combinations $S^j = \{s_1^j, \ldots, s_N^j\}$, where $(s_1^j \in S_1), \ldots, (s_N^j \in S_N)$ do
      calculate the delay using Equation 2.3
      normalize the delay w.r.t first strategy combination
      calculate the crosstalk noise using Equation 2.7
      normalize the noise w.r.t first strategy combination
      $P[i,\, S^j] \leftarrow$ Geometric mean of normalized noise and delay
    end for
  end for
  return: payoff matrix $P$ $\forall$ strategy combinations
```

The two most affecting aggressor nets are chosen among its pool of aggressors based on the fraction of its contribution to the score of the given net. The two aggressor nets, a_1 and a_2, which contribute to the majority of the score value are selected. The wires i, a_1 and a_2 are added to the set of strategy determiners and represent the set of scalable wires in the game. The driver and receiver gates of wires i, a_1 and a_2 are added to the set of strategy determiners as the set of scalable gates (a total of six gates). The minimum wire size of a net is fixed based on the minimum wire size design rule requirement of the process technology. The maximum wire size for a net is determined based on the free space available around the net. The range between maximum and minimum wire sizes for each net is treated as its possible wire sizes without violating the process design rules. This range is divided into a discrete set of values with equal step sizes and represented as the set of scalable wire sizes for the corresponding wires. The scalable gate sizes for each gate are chosen such that its replacement in the design do not result in rerouting. The maximum scalable gate size depends on two factors: (i) available free space surrounding the gate in the design, and (ii) drive strengths available for a gate type in the standard cell library. For a gate at specific location in the design, all the gate sizes supported by the standard cell library cannot be used as its scalable sizes. Some of the gate sizes cannot be fitted within the available free space without disturbing the routed nets surrounding it. Hence, the number of scalable sizes available for each gate is always less than or equal to the number of drive strengths available in the standard cell library. The collection of the scalable wire sizes of the three wires and the scalable gate sizes of the six gates are modeled as the strategy set of each player. If a designer wants to give weight to a particular design parameter (say crosstalk noise), then the strategy set can be pruned to have scalable sizes which are favorable to that parameter. The payoffs for two player are calculated by using the Equations 2.3 and 2.7 respectively, according to the actions chosen by the players.

Algorithm 6.3. Algorithm for Nash Equilibrium Solution

```
Input: Number of players $N$, Payoff matrix $P$
Output: Nash solution
  for all players $i$ do
    for all payoffs of player $i$ do
      find $s_i^*$ such that,
      $P_i(s_1^*, \ldots, s_i^*, \ldots, s_N^*) \geq P_i(s_1^*, \ldots, s_i, \ldots, s_N^*)$
      $s_i^*$ is the Nash strategy for player $i$
    end for
  end for
  Nash-solution $S^* = \{s_1^*, \ldots, s_N^*\}$
     % set of optimized strategies for all $N$ players
  return: Nash solution $S^*$
```

The Nash equilibrium is evaluated for the chosen net, i, and the game is played out. The six gates and three wires participated in the played game are flagged as “sized” and their sizes are set equal to the calculated Nash sizes. The nets i, a_1 and a_2 are tagged as played and are removed from the sorted list of ordered nets. A new net located at the head of the sorted list is selected to play the next game. The six gates and three wires corresponding to the selected net are identified to participate in the newly formed 2-player game. The scalable sizes of the gates which are marked as sized are assigned to a singleton consisting of only its calculated Nash width. The scalable sizes for unmarked elements in the set of strategy determiners are identified as described above and the Nash equilibrium of the game is evaluated. This process of creating and playing the sequential games is repeated until the sorted list of wires is empty. It can be noted that three wires are removed from the sorted list for every game played and hence, the total number of games played is equal to $N_{wires}/3$, where N_{wires} is the total number of wires in the design. The Algorithm 6.1. shows the pseudo-code of the integrated gate and wire sizing algorithm developed in this work.

6.3.2 Approach 2: Gates Ordered Based on Delay Criticality

The ordering of nets in the sorted list dictates the order in which the gates are considered for their size optimization. In section 6.3.1, the interconnect wires are sorted in a list based on the noise criticality of the nets. Hence, the approach outlined in section 6.3.1, yields slightly better optimization of crosstalk noise than interconnect delay, while simultaneously optimizing both delay and noise. In this section, we investigate a strategy wherein delay is considered as higher criticality than noise, while simultaneously optimizing delay and noise. It is interesting to note that both methods yield significantly better optimization of both delay and noise compared to other methods. The designer can choose either of the strategies based on the need. The difference between the two strategies is the way in which the sorted list is created. After the design is placed and routed, the path delays of all the paths in the design are estimated, and are sorted into a list based on their delay criticality. The most delay critical path is chosen to create games for the gate and wire size optimization. The games are created for each net in the chosen path in the order from its primary output to primary inputs. As an example for illustration, consider the chosen path to consist of four gates: A, B, C and D, in successive transition connected with nets: $1, 2$ and 3, respectively. The gate A is driven by primary inputs and gate D drives a primary output. In order to consider the down-stream load capacitance, the net 3 connecting gates C and D should be optimized before the nets 1 and 2. Thus, the games are played in the order of net 3 followed by net 2 followed by net 1. The game formulated for each net is a two player game with the interconnect delay and crosstalk noise as players. The two most aggressor nets and the corresponding strategy values are identified as indicated in section 6.3.1. After the games are played for all the nets of the chosen critical path in its direction of primary output to primary inputs, the next critical path in the sorted list is selected to play the games. This process of creating games is repeated until the sorted list is empty.

6.4 Time and Space Complexity

The worst case time complexity of evaluating Nash equilibrium for a general M-player game with S strategies for each player is given as $O(M * S^M)$ [52]. Referring to Section 6.3.1, we have modeled the problem of gate and wire sizing for simultaneous interconnect delay and crosstalk noise reduction as a game with two players. For each gate type, the number of different drive strengths available for it in the standard cell library act as its maximum possible gate sizes. We have used a standard cell library containing gates with four different drive strengths, build on TSMC 180nm design rules. We have divided the available space between wires such that the wires involved in the game has a maximum of five different possible wire sizes. The actual scalable sizes for each gate and wire depends on its location and the free space surrounding it in the design. Hence, the number of scalable sizes available for each gate and wire is always less than or equal its maximum possible sizes. Each game involves six gates and three wires. Hence, the number of strategies for each player is given by $S_p \leq (6 * 4 + 3 * 5)$. Thus, the worst case complexity of calculating the Nash equilibrium for a single game played is given by $O(2 * S_p^2)$. The games are played repeatedly until the sorted list is empty. Initially, the sorted list consists of all the wires of the design. After a game is played, three wires are tagged as “sized” and removed the list. Hence, the total number of games played is given by $N_{wires}/3$. Thus, the overall worst case time complexity of the proposed integrated algorithm can be given mathematically as

$$O\left(\frac{N_{wires}}{3} * 2 * S_p^2\right) + O\left(N_{gates}\right) \approx O\left(N_{wires}\right) + O\left(N_{gates}\right)$$

where N_{wires} is the total number of wires and N_{gates} is the total number of gates in the given design. It can be noticed that the time complexity of the proposed algorithm is *linear* and is proportional to the total number of gates and wires of the design. The space complexity of the proposed algorithm is dependent entirely on the number of wires in the design and the space complexity of payoff matrix. The space complexity of a payoff matrix depends on the number of strategies for each player playing the game. As the games are played sequentially, the total space required by all the games is equal to the space complexity of a game involving players with maximum number of strategies. Mathematically, the space complexity required by all the payoff matrices is given as $O(6*4+3*5)$. Hence, the space complexity of the proposed algorithm is given as

$$O\left(N_{wires}+6*4+3*5\right) \approx O\left(N_{wires}\right)$$

6.5 Proof of Existence of Nash Equilibrium for the Proposed Integrated Gate and Wire Sizing Formulation

In this section, we provide the proof of existence of Nash equilibrium in the case of gate and wire sizing problem for simultaneous optimization of interconnect delay and crosstalk noise. As the gate and wire sizes increases, the interconnect delay of the net decreases and the coupling capacitance increases resulting in a convex payoff function. In this work, we have modeled the interconnect delay and crosstalk noise of the chosen net as the players of the game. Let $G = \{S_1, \ldots, S_n; f_1, \ldots, f_n\}$ be a game with each player $i \in N$ having a strategy set S_i containing the scalable gate and wire sizes, and its payoff given by f_i. We have modeled the strategy set S_i for each player as a non-empty, compact set of a finite dimensional Euclidean space. Because of the convex nature of the interconnect delay and crosstalk noise, the modeled payoff function f_i becomes upper semicontinuous on $S = \Pi_{i=1}^{N} S_i$ and for any fixed $u_i \in S_i$, the function $f_i(u_i, .)$ is a lower semicontinuous on $S_{(-i)}$ [16]. For any $u \in S$, the best reply or the expected payoff $B_i(u)$ is also convex. According to Kakutani's fixed point theorem [61], the game G has at least one Nash equilibrium point if the graph

$$G_B = \{(x, y) : x \in S, y \in B(x)\}$$

is closed. Lets assume that its not closed. Then, $\exists(x^0, y^0) \notin G_B$, such that every neighborhood (in $S \times S$) of (x^0, y^0) contains a point of G_B.

$\because x^0$ is a gate or wire size, it has to be one of those from the set of possible gate and wire sizes for the given player in order to satisfy the DRC rules of the used process technology.

$\therefore x^0 \in S \Rightarrow y^0 \notin B(x^0)$

In other words, for at least one player playing the game (say crosstalk noise), there is an $y_1^1 \in S_1$ such that

$$f_1(y_1^1, x_2^0, \ldots x_n^0) \geq f_1(y_1^0, x_2^0, \ldots x_n^0) \tag{6.1}$$

Let F be a function such that $F : S^2 \rightarrow \Re$ and given as

$$F(x, y) = f_1(y_1^1, x_2, \ldots x_n) - f_1(y_1, x_2, \ldots x_n)$$

Since f_i is upper semicontinuous on S and $f_i(u_i, .)$ is lower semicontinuous on S_{-i}, F is lower semicontinuous and $C = \{(x, y) \in S^2 : F(x, y) \leq 0\}$ is closed. Hence, for any $(\bar{x}, \bar{y}) \in G_B, F(\bar{x}, \bar{y}) \leq 0$. But, by Equation 6.1, $F(x^0, y^0) \geq 0$, contradicting the closedness of C. Thus, there is a point $s^* \in S$ such that $s^* \in B(s^*)$, which is a Nash equilibrium point.

6.6 Design Flow

The design flow for obtaining an optimally gate and wire sized circuit from a verilog/VHDL description is shown in Figure 6.1.. The behavioral verilog/VHDL description is synthesized on to a library of standard cells and given as input to the design flow. The standard cells are placed and routed in coherence with the synthesized code using any standard cell place and route tool. We have used the First Encounter™ RTL-to-GDSII tool from Cadence® Design Systems to perform the placement and routing of gate-level

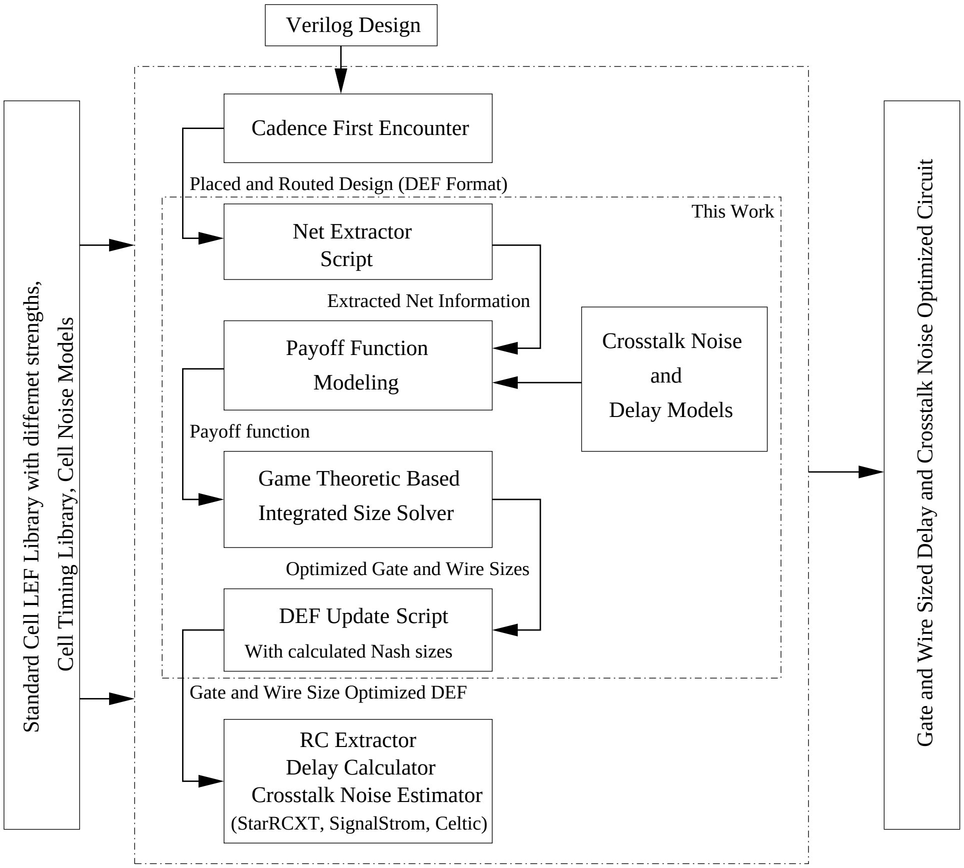

Verilog Design
Cadence First Encounter
Placed and Routed Design (DEF Format)
This Work
Net Extractor Script
Extracted Net Information
Payoff Function Modeling
Crosstalk Noise and Delay Models
Payoff function
Game Theoretic Based Integrated Size Solver
Optimized Gate and Wire Sizes
DEF Update Script
With calculated Nash sizes
Gate and Wire Size Optimized DEF
RC Extractor
Delay Calculator
Crosstalk Noise Estimator
(StarRCXT, SignalStrom, Celtic)
Standard Cell LEF Library with differnet strengths, Cell Timing Library, Cell Noise Models
Gate and Wire Sized Delay and Crosstalk Noise Optimized Circuit

Figure 6.1. Integration of Proposed Gate and Wire Sizing Algorithm in the Design Flow

RTL design. The parasitics from the routed design are exported in SPEF format with the help of StarRCXT from Synopsys Inc. A lex and yacc script is developed to read the SPEF netlist and extract the values of interconnect resistance, capacitance, inductance, along with their set of aggressor gates for all the nets of the design. Also, a gawk script is written to extracts the information about the length of overlaps between two given nets along with their distance of separation from the DEF netlist. The payoff function is used by the proposed integrated size solver described in Algorithm 6.1. to minimize the interconnect delay and crosstalk noise of individual nets in the order of their noise criticality or delay criticality. The optimized gate and wire sizes resulted from our solver are used to update the routed design. We have developed another gawk script which scales the gates and wires in the original DEF routed design according to their calculated optimized sizes.

It should be noted here that the resulting optimized design does not require rerouting since the possible gate and wire sizes drafted are within its limits to satisfy the design rules of the used process technology.

6.7 Experimental Results

The game theoretic gate and wire size solver described in Algorithm 6.1. was implemented in C and executed on a UltraSPARC-IIe 650MHz, 512MB Sun Blade 150 system running Solaris 2.8 on it. The ASIC designs on which we tested our algorithm were obtained from Opencores [62]. A standard cell library containing 10 logic cells with 4 different drive strengths based on a 6-Metal layer, 180nm technology has been developed and used. ASIC designs are mapped to the standard cells using BuildGates, an RTL synthesis tool of Cadence design systems. We have modified the designs such that the blocks are flattened to standard cells without maintaining the hierarchy. The on-chip memory modules are realized as register arrays with D-flipflop as basic building units. The structural design is inputed to Cadence First Encounter to perform the placement and routing. The information needed for the game theoretic solver is extracted from DEF and SPEF files using a lex and yacc script. The calculated gate and wire sizes from the game theoretic solver is used to update the original DEF file in order to generate an optimized DEF file. It can be noted that the optimized DEF file is created with the help of gawk script and verified for the DRC rules. The design is not rerouted to generate the optimized DEF file. The interconnect delay and crosstalk noise are estimated using Cadence Signalstorm and CelticIC respectively with their robust models, and not using the analytical models used during optimization.

Table 6.1. Average Savings for Simultaneous Optimization of Interconnect Delay and Crosstalk Noise During Integrated Gate and Wire Sizing *

Open Core Design [62]	Total Gates	Total Nets	WS-GT [1]			GS-GT [2]			Integrated-SA [3]			Integrated-GT-NC [4]			Integrated GT-DC [5]		
			Run Time (mins)	% Average Savings		Run Time (mins)	% Average Savings		Run Time (mins)	% Average Savings		Run Time (mins)	% Average Savings		Run Time (mins)	% Average Savings	
				Delay	Noise		Delay	Noise		Delay	Noise		Delay	Noise		Delay	Noise
Mult	428	854	1.89	9.87	12.13	1.13	4.18	5.34	21.07	7.48	4.91	4.05	14.80	18.02	3.79	16.76	14.51
PCI	7882	19520	5.23	19.32	21.42	4.29	10.39	16.49	64.91	14.20	15.82	8.13	26.71	31.95	8.36	34.12	28.59
ATA	21781	43563	11.86	29.87	20.14	10.49	20.31	23.91	192.53	26.91	19.25	13.92	38.91	36.12	14.33	43.57	35.18
RISC	34172	61468	16.86	25.22	22.31	13.31	15.28	14.05	260.16	23.97	16.01	18.34	30.04	29.56	17.58	34.68	26.91
AVRμP	41274	78770	21.32	22.45	31.34	15.79	21.41	22.96	428.91	19.78	25.83	24.87	29.74	42.67	25.84	35.16	36.78
P16C55	52128	102021	28.98	19.86	43.29	19.98	16.96	21.91	514.04	14.19	34.89	32.19	24.07	44.26	31.92	27.48	43.02
T80μC	69973	157850	39.48	23.78	33.12	27.67	19.86	25.24	774.38	20.23	29.37	43.16	28.38	35.49	45.27	31.83	31.97
Average										18.11	20.87		27.52	34.03		31.94	30.99

* No area overhead incurred for any of the approaches. The percentage values indicated are w.r.t placed and routed design without gate and wire sizing.

[1] WS-GT: Game theoretic wire sizing approach for simultaneous optimization of interconnect delay and crosstalk noise

[2] GS-GT: Game theoretic gate sizing approach for simultaneous optimization of interconnect delay and crosstalk noise

[3] Integrated-SA: Integrated simulated annealing gate and wire sizing approach for simultaneous optimization of interconnect delay and crosstalk noise

[4] Integrated-GT-NC: Integrated game theoretic gate and wire sizing approach for simultaneous optimization of interconnect delay and crosstalk noise with nets ordered based on noise criticality

[5] Integrated-GT-DC: Integrated game theoretic gate and wire sizing approach for simultaneous optimization of interconnect delay and crosstalk noise with gates ordered based on delay criticality

Table Legend: *Avg*: Average savings of all the nets in the entire design; *Crit*: Savings on the critical net of the design; *Runtime*: running time of each algorithm.

Table 6.2. Critical Path Savings for Simultaneous Optimization of Interconnect Delay and Crosstalk Noise During Integrated Gate and Wire Sizing

Open Core Design [62]	Total Gates	Total Nets	WS-GT			GS-GT			Integrated-SA			Integrated-GT-NC			Integrated GT-DC		
			Run Time (mins)	% Crit. Path Savings		Run Time (mins)	% Crit. Path Savings		Run Time (mins)	% Crit. Path Savings		Run Time (mins)	% Crit. Path Savings		Run Time (mins)	% Crit. Path Savings	
				Delay	Noise		Delay	Noise		Delay	Noise		Delay	Noise		Delay	Noise
Mult	428	854	1.89	11.79	17.23	1.13	5.59	7.86	21.07	8.56	9.16	4.05	17.19	17.99	3.79	20.11	15.27
PCI	7882	19520	5.23	34.21	37.38	4.29	17.11	25.29	64.91	19.96	20.63	8.13	40.03	45.08	8.36	41.64	39.35
ATA	21781	43563	11.86	39.95	42.15	10.49	24.78	27.12	192.53	23.18	20.37	13.92	41.61	41.16	14.33	48.49	38.92
RISC	34172	61468	16.86	35.21	29.73	13.31	21.16	19.96	260.16	42.16	22.07	18.34	47.94	37.41	17.58	51.42	32.13
AVRμP	41274	78770	21.32	37.63	40.31	15.79	27.19	28.95	428.91	28.92	36.10	24.87	39.82	53.58	25.84	43.51	40.10
P16C55	52128	102021	28.98	27.45	57.98	19.98	21.54	23.16	514.04	21.61	35.92	32.19	26.88	57.74	31.92	29.15	53.58
T80μC	69973	157850	39.48	34.87	39.89	27.67	24.21	28.92	774.38	22.84	34.57	43.16	37.33	44.23	45.27	43.08	39.76
Average										23.89	25.55		35.83	42.46		39.63	37.02

The notations used in this table are same as in Table 6.1.. *Crit. Path Savings*: Savings on the critical net of the design

The works reported in literature solve the problem of gate and wire sizing for crosstalk noise optimization under delay constraints. In this work, we have solved the problem of integrated gate and wire sizing for simultaneous optimization of crosstalk noise and interconnect delay. Hence, to compare our results, we have implemented simulated annealing for simultaneous optimization of crosstalk noise and interconnect delay, and executed it on same Solaris machine with same set of inputs and parameters. The implemented simulated annealing algorithm is experimented to obtain the best results in terms of optimization of interconnect delay and crosstalk noise. The set of possible gate and wire sizes are calculated as indicated in Section 6.3.1. In each move of simulated annealing process, a gate or a wire is randomly selected and its size is randomly assigned from the set of its possible sizes. The cost function is defined as the geometric mean of interconnect delay and crosstalk noise summed over all the nets. The initial temperature of simulated annealing process is determined by finding the average change in the cost for a set of random moves from the starting configuration and selecting the temperature which leads to an accept probability of 0.95. The number of moves per temperature for each design is set to 20 times the total number of nets and gates in the design. This is done so as to allow at least 10 to 15 moves on an average for each gate or net before settling for its solution. The up-hill moves are accepted with a probability of $e^{-\frac{\delta C}{T}}$, where δC is the change in the cost and T is the current temperature of the iteration. The temperature is cooled at the rate of 0.95.

The experiments were conducted such that the area overhead is zero in both simulated annealing and game theoretic approaches. Referring to Tables 6.1. and 6.2., simulated annealing shows an average improvements of 18.11% and 20.87% in terms of interconnect delay and crosstalk noise respectively. When the games are ordered according to the noise criticality of the nets, game theoretic approach shows an average improvements of 27.52% and 34.03% in terms of interconnect delay and crosstalk noise respectively. Whereas, when the games are ordered according to the delay criticality of the paths, game theoretic approach shows an average improvements of 31.94% and 30.99% in terms of interconnect delay and crosstalk noise respectively. Hence, games ordered according to noise criticality results in better noise optimization, while games ordered according to delay criticality results in better delay optimization. Thus, the designer can choose either of the proposed strategies based on the need of the design. Game theoretic integrated size solver based on both noise and delay critical strategies, in addition to outperforming simulating annealing in terms of interconnect delay and crosstalk noise savings, has significantly smaller runtimes. *Hence, our approach is scalable and can handle the complexity of large SOC designs.*

Table 6.3. Crosstalk Noise Optimization Under Delay Constraints During Integrated Gate and Wire Sizing

Open Core Design [62]	Number of Noise Violations Noise Threshold = $0.15V_{dd}$	
	Lagrangian [31]	Game [This Work]
Mult	7	5
PCI	18	7
ATA	63	16
RISC	121	33
AVRμP	167	47
P16C55 μC	204	58
T80μC	257	71

To enable a direct comparison of our work with one of the recent works developed in [31], we have modified our game theoretic approach to minimize the crosstalk noise under delay constraints. The work developed in [31] is a Lagrangian relaxation based gate sizing approach for reducing the crosstalk noise under the delay constraints. We have extended the Lagrangian gate sizing approach reported in [31] based on the work of [24], which is also Lagrangian based, in order to include the wire sizing along with gate sizing. For both game theoretic and Lagrangian relaxation approaches, we have used the delay values obtained from simultaneous optimization of interconnect delay and crosstalk noise as the set of delay constraints, so as to ensure a tighter constraint set for both approaches. The Lagrangian relaxation problem for post-layout crosstalk noise optimization is solved using LANCELOT [66]. In game theoretic approach, the strategies which do not satisfy the delay constraints are pruned out from their respective strategy sets. The game is played with the three nets, namely, the chosen critical net and its two most aggressor nets, as the players of the game. The payoff function is determined by the crosstalk noise induced on the chosen net. Table 6.3. shows the comparison of game theoretic and Lagrangian relaxation based approaches indicated in terms of number of noise violations for each design. Noise violations are expressed as the number of nets which have an induced noise exceeding a threshold noise set to $0.15V_{dd}$. It can seen that our approach results in significantly less number of faults when compared to the Lagrangian relaxation.

6.8 Conclusions

Game theory allows the simultaneous optimization of multiple metrics in the context of conflicting objectives leading to a convex objective function in the problem formulation. This essentially makes it possible to use game theory for simultaneous optimization of interconnect delay and crosstalk noise. Optimizing both interconnect delay and crosstalk noise is extremely critical in deep submicron and nano regime circuits. We have developed an integrated framework for performing both gate and wire sizing simultaneously on any given design. The developed integrated framework performs significantly better than sequential application of gate sizing followed by wire sizing. Also, the proposed method results in a linear time algorithm with significantly better results than simulated annealing and Lagrangian relaxation, making this work an important contribution.

CHAPTER 7

STATISTICAL GATE SIZING UNDER PROCESS VARIATIONS

In this chapter, we develop a new post-layout gate sizing algorithm for simultaneous reduction of delay uncertainty and crosstalk noise under the impact of process variations. The problem of post-layout statistical gate sizing is modeled as a 2-player stochastic game and solved using Nash equilibrium theory. The delay uncertainty induced on a net depends on the size of its driver and receiver. The crosstalk noise induced on a net depends on the size of its driver and the sizes of the gates driving its coupled nets. Increasing the size of a gate will affect the sizes of other related gates and result in a conflicting situation, thus, making the gate sizing a challenging problem. In addition, due to process variations, the gate sizes are no longer deterministic, but rather behave as a probabilistic distribution over a range. Stochastic games allow the modeling of probabilistic distribution of gate size space and also effectively capture the conflicting nature of the problem. We have implemented two different strategies in which the games are ordered according to (i) the noise criticality, and (ii) the delay criticality of nets.

In Chapter 5, we had solved the gate sizing problem for simultaneous optimization of interconnect delay and crosstalk noise using four-player games, without considering process variations. The developed algorithm is shown to be significantly better than non-linear optimization using Lagrangian multipliers, simulated annealing and genetic search, in terms of optimization and runtimes. In this chapter, we have formulated the gate sizing problem as a two-player stochastic game for simultaneous minimization of delay uncertainty and crosstalk noise under the impact of process variations. We have also considered the spatial correlations due to process variations. The modeling of games in this work is completely different from that given in Chapter 5. Here, we have modeled the delay uncertainty and crosstalk noise as the players of the 2-player stochastic game.

7.1 Problem Definition

The problem of post-layout gate sizing under process variations can be defined as follows: find the optimal gate sizes under the impact of process variations such that the delay uncertainty and crosstalk noise of the overall circuit is minimized under the given area constraints and without the need for rerouting any of the nets in the design. The coupling capacitance of a net depends on its wire size, the length of overlap and the spacing between adjacent nets. This information can be efficiently extracted at post-routing phase. The coupling noise induced on a net depends on the size of its driver, receiver and aggressor gates. Also, the delay uncertainty is a function of the gate sizes, and the input and load capacitances. Equations 2.6 and 2.7 emphasize that the gate sizes directly control the delay uncertainty and crosstalk noise in terms of the driver resistances, gate and coupling capacitances. Hence, calculating the optimal gate sizes can effectively reduce the crosstalk noise and delay uncertainty in deep submicron designs. The deterministic method optimizes only the critical and the near critical nets of the design and does not improve the non-critical paths. Due to the intra-die variability, non-critical paths become critical causing the statistical circuit delay and noise to deteriorate [67]. Hence, we have considered all the nets of the design for minimization of their delay uncertainty and crosstalk noise using a stochastic framework. Gate sizing can be performed at post-route level by utilizing the existing fill-space. In this work, we incrementally scale the gate sizes to utilize the available fill-space such that the routed resources in adjacent regions are not disturbed. Hence, our approach will neither result in area overhead nor need re-routing of the design.

7.2 Motivation for Statistical Gate Sizing Problem

Traditionally, in VLSI design, a single parameter is optimized assuming other parameters as constraints. In deep submicron and nanometer circuits, the simultaneous optimization of power, delay and crosstalk noise is becoming important. Further, the optimization methods need to take into account the impact of process variations arising from the fabrication process. The optimization in a multi-metric and conflicting environments is a difficult problem since the normal definition of an optimal value no longer applies and valid. For example, an optimal gate size for one metric may not be optimal for another metric. The optimal policy at any given instance depends on the polices for other metrics keeping the best interest of the entire system in view. While most optimization methods such as ILP, simulated annealing, and force directed methods lend themselves well for single metric optimization, these methods are inadequate for multi-metric optimization. Thus, there is a need for new methods and algorithms to be developed which applies the multi-agent optimization theories to the problems of VLSI CAD. The consideration of process variations during the design optimization requires probabilistic analysis due to the uncertainty element introduced by process variations. Most of the previous works reported in literature employs a deterministic framework to perform gate sizing by using either statistical static timing analysis or by developing a statistical gate delay model (please refer to Table 2.1.). In [68], it is pointed out that a stochastic framework is needed to analyze the systems with statistical parameters since the system itself is an outcome of a stochastic process. This motivates us to develop a stochastic framework for multi-metric optimization in order to perform gate sizing under the impact of process variations.

The fundamental basis and structure of game theory and stochastic games allows the formulation of optimization problems in which multiple inter-related cost metrics compete against one another for their simultaneous optimization. Further, stochastic games inherently captures the nondeterministic behavior of the system parameters. Game theoretic reasoning takes into account the attempts made by the multiple agents towards the optimization of their objectives for every decision. Each agent or a player's decision is based on the decision of every other player in the game and hence it is possible for each player to optimize his gain with respect to the others' gains in the game. In terms of game theory, a solution is said to reach its global value for the given conditions when it reaches its equilibrium. Stochastic games are played in stages, where the payoff values for each player and the state transition from one stage to another are controlled by a stochastic function. This enables the designer to efficiently capture the uncertainty due to process variations with the help of stochastic function and control the optimization at each stage of the game using its probability distribution. Thus, game theory ideally suits for multi-metric optimization with conflicting objectives and the stochastic function of the stochastic games can be used to model the uncertainty in design parameters due to process variations. Further, if the payoff function is convex, i.e., the parameters being optimized correspond to conflicting objectives (such as delay uncertainty and crosstalk noise in this work), game theoretic optimization, in fact, performs significantly better [18].

7.3 Proposed Statistical Gate Sizing

In this section, we present the stochastic game theoretic methodology for minimizing the delay uncertainty and crosstalk noise. Again, we have developed two different strategies based on the ordering of the games. The two orderings are based on the noise criticality or the delay criticality of the gates and each strategy is discussed in detail below.

7.3.1 Approach 1: Gates Ordered Based on Noise Criticality

In this work, we perform the optimization task after the design is placed and routed. The interconnect resistance, capacitance, inductance, and coupling capacitances along with their aggressor drivers are extracted for each net from the SPEF netlist of the routed design. The length of interconnect wires and the length of overlap of each net with its set of aggressor nets and their spacing are extracted from the routed design exported in DEF format. A multi-terminal net is assumed as different nets with same driver and different receivers. The gates of the design are ordered in accordance to the noise criticality of its driven nets. The use of a statistical static timing and crosstalk noise analyzer to estimate and identify the noise critical nets is time consuming. In addition, statistical static analysis makes assumptions of normal distributions for the signal arrival time and the slope, and approximates the resultant of two or more normal distributions as normal distribution, thereby, leading to inaccurate analysis [37]. Hence, in this work, we propose a heuristic to rank the nets relative to each other to indicate whether a net is more noise critical than other nets or vice versa, rather than estimating the noise on each net. The coupling capacitance between two nets is proportional to the length of overlap and inversely proportional to the square of their spacing [57]. Hence, we define the rank for each net as

$$Rankofanet = \sum_{\forall aggressors} \frac{\text{(Length of overlap)}}{\text{(Spacing)}^2}.$$

The nets are sorted and arranged in a list according to their rank. The most critical net will form the head of the list and the least critical net will form the tail. The gates influencing the critical noise nets are considered first for their optimization. For any given net, there can be many potential aggressor nets. It is indicated in [10] that it is virtually not possible to consider all the aggressor gates in the coupled set of a net. In our work, we have used crosstalk noise models which considers the effects of two aggressor nets on the victim net. The first two aggressor nets which contribute to the major fraction of the rank of given net are marked as its most affecting aggressor nets, and are used while sizing the victim net. The limitation to two aggressor nets is not due to the modeling used in this work, but due to the crosstalk noise models used. The crosstalk noise models have to replaced with a more accurate higher order models in order to consider three or above aggressor nets while sizing any victim net. For each gate in the design, we have chosen its gate sizes such that their replacement in the design does not result in re-routing. The set of various possible gate sizes with which it can be scaled without violating DRC rules are stored as its scalable size set. The maximum gate size with which a gate can be sized depends on (i) the available gate sizes for its gate type in the standard cell library and (ii) the existing free-space around the given gate in the design. The existing free-space is partitioned among the gates such that it can be replaced without disturbing the

Algorithm 7.1. Post-Route Gate Sizing Algorithm for Simultaneous Optimization of Delay Uncertainty and Crosstalk Noise under Process Variations

```
Input: Placed and routed design
Output: Optimized gate sizes
  - extract the net parasitics from SPEF file
  for all gates do
    determine_aggressors();
      % extract aggressor gates from SPEF file
    - mark the gate as un-played
  end for
  for all nets do
    - extract the overlapping lengths and spacing between the adjacent nets from DEF file
    calculate_scores();
    sort_scores();
  end for
  while there exists an unsized gate do
    - select an un-played noise critical net $i$ from the sorted list
    - identify two main aggressor gates for net $i$ in the order of their coupling effects
    - create a 2-player game for delay uncertainty and crosstalk noise
    - the participating elements for determining strategy set of the two players are the
      driver, the receiver, and the two main aggressor gates of net $i$
    for gate $g_k$ among the four identified gates do
      if $g_k$ is marked as sized then
        scalable_size_set of $g_k$ $\leftarrow$ calculated_best_gate_size;
      else
        scalable_size_set of $g_k$ $\leftarrow$ determine_strategies();
      end if
    end for
    - represent the scalable_size_set of four gates as the strategy set of both players, rep-
      resented as $(s)$
    Nash_sizes $\leftarrow$ Sparse_Game($s, T$);
      % for Sparse game algorithm, please refer to Algorithm 7.2.
    - mark these four gates as sized
    - mark the net $i$ as played
  end while
  return: optimized Nash sizes of gates
```

neighboring routes and satisfies DRC rules. The minimum gate size for each gate is chosen as the size of minimum strength gate available in the standard cell library for its gate type.

A two-player nonzero-sum stochastic game is formed for the most critical net available at the head of the sorted list. The delay uncertainty and crosstalk noise act as the two players of the game. The driver, the receiver and the two most affecting aggressor gates of the selected noise critical net are identified to participate in the game. The scalable size set of these four gates put together form the strategy sets of the two players. If a gate is marked as sized, its scalable size is a singleton containing its best size calculated from previously played-out stochastic game. The stochastic game is played in stages, with each stage represented as a one-shot normal form game. We have used finite-horizon undiscounted stochastic games in our modeling since these games stop after a fixed number (T) of stages. At each stage, the players choose an strategy (gate sizes for the four participating gates) which is more beneficial to him.

Due to the process variations, the gate sizes are no longer deterministic quantities, but behave as random variables around their nominal values. As predicted in [4], we have used a variation of 25% around the nominal value in gate sizes due to process variations. This variation in the gate size for a gate g_k is captured using a probability distribution $P(g_k)$. The proposed gate sizing methodology does not make any assumptions about the distribution of parameter variations. The widely used parameter distributions are Gaussian or Normal and Log-Normal distributions. The strategy of a player is a vector containing a size assignment to each of the four gates participating in the game. For example, if scalable size set of $g_1 = \{g_{11}, g_{12}, g_{13}\}; g_2 = \{g_{21}, g_{22}\}; g_3 = \{g_{31}, g_{32}\}; g_4 = \{g_{41}, g_{42}, g_{43}, g_{44}\}$, then a possible strategy for a player can be $s' = \{g_{11}, g_{21}, g_{31}, g_{41}\}$, another possible strategy $s'' = \{g_{12}, g_{21}, g_{32}, g_{44}\}$ and so on. Hence, there will be a large set of strategies for each player, resulting in a large state space. The "sparse sampling algorithm" developed by Kearns et al [69] is modified to solve the formulated 2-player stochastic game (please refer to Algorithm 7.2.). The modifications are made in order to consider the spatial correlations due to process variations and to include delay uncertainty and crosstalk noise as the players. The interesting aspect of this algorithm is that its time complexity is independent of the size of the state space, thereby making our methodology tractable in spite of having a large state space. The algorithm is a recursive algorithm which takes any state s and time T as the inputs. It assumes access to an arbitrary fixed Nash selection function f, generates immediate payoff matrices $M_k[s]$, and samples the probability distribution $P(g_k)$. We have

Algorithm 7.2. Sparse Game Algorithm for Computing Approximate Nash Equilibrium in Stochastic Games

```
*Function Name:* Sparse_Game($s, T$)
*Input:* Strategy set of both players, Number of steps $T$
*Output:* Nash equilibrium policy: $(\alpha, \beta)$
  if $T = 0$ then
    $M_1[s] \leftarrow$ calculate crosstalk noise using Equation 2.7
    $M_2[s] \leftarrow$ calculate delay uncertainty using Equation 2.6
      % $s$ represents the set containing the nominal gate size values
    $\alpha \leftarrow f_1(M_1[s], M_2[s])$
    $\beta \leftarrow f_2(M_1[s], M_2[s])$
      % $f_1, f_2$ are arbitrary Nash selection functions
    for $k \in \{1, 2\}$ do
      $Q_k \leftarrow M_k[s](\alpha, \beta)$
    end for
    *return:* $(\alpha, \beta, Q_1, Q_2)$
  end if
  for all strategy pair $(i, j) \in s$ do
    - select a gate $g_k$ at random
    - select $m$ random samples $s'_1, \ldots, s'_m$ from $P(.|g_k, i, j)$
      %$P(g_k)$ represents the probability distribution of the gate
        sizes around its nominal values due to process variations.
    for l $= 1, \ldots, m$ do
      - transit the gate sizes of remaining three gates in the sample $s_l$ according to the
      spatial correlations in gate size variations
      $(\alpha', \beta', Q_1[s'_l, T-1], Q_2[s'_l, T-1]) \leftarrow Sparse\_Game(s'_l, T-1)$;
    end for
    for $k \in \{1, 2\}$ do
      $Q_k[s, T](i, j) \leftarrow M_k[s] + (1/m) \sum_{l=1}^{m} Q_k[s'_l, T-1]$
    end for
  end for
  $\alpha \leftarrow f_1(Q_1[s, T], Q_2[s, T])$
  $\beta \leftarrow f_2(Q_1[s, T], Q_2[s, T])$
  for $k \in \{1, 2\}$ do
    $Q_k \leftarrow Q_k[s, T](\alpha, \beta)$
  end for
  *return:* $(\alpha, \beta, Q_1, Q_2)$
```

used the Nash selection function as to return the minimum payoff for the player under the given strategy of the other player.

The spatial correlations in gate sizes due to process variations are modeled exactly as given in [37]. The gate sizes of devices located in same grid are assigned perfect correlations, gate sizes in neighboring grids are assigned high correlations and zero correlations are assigned between other gates. At each stage of the stochastic game, m samples are taken from the probability distribution of gate size space $P(g_k)$ of a randomly selected gate g_k. The gate sizes for the remaining gates of the m selected samples are scaled according to their spatial correlation with respect to gate g_k. The payoff matrices are updated according to the selected m samples and hence, the game moves to the next stage depending on the probability distribution of the process variations. It has been proved in [69] that the number of samples m required at each stage for obtaining a near-Nash solution is given by $m > (T^3/\epsilon^2)\log(T/\epsilon) + T\log(n/\epsilon)$, where $\epsilon > 0$ and n is the number of strategies available to both players at any stage.

In our work, we have used T as 10 and m as given by the above inequality. The game makes transitions to new stages until T steps are reached. This is the stopping criterion for the two player nonzero-sum stochastic game. The time complexity of finding the Nash equilibrium given in Algorithm 7.2. is proportional to m^T, due to the recursion for T stages. After the 2-player stochastic game is played out, the involved four gates are marked as sized and the selected noise critical net is removed from the sorted list. The next noise critical net available at the head of the sorted list is selected to create a new 2-player stochastic game as illustrated above and the game is played out. This process of formation of 2-player stochastic games continues until all the gates of the design are marked as sized. Algorithm 7.1. represents the pseudo-code of the proposed gate sizing algorithm for simultaneous optimization of delay uncertainty and crosstalk noise under process variations.

7.3.2 Approach 2: Gates Ordered Based on Delay Criticality

The ordering of nets in the sorted list dictates the order in which the gates are considered for their size optimization. In section 7.3.1, the interconnect wires are sorted in a list based on the noise criticality of the nets. Hence, the approach outlined in section 7.3.1, yields slightly better optimization of crosstalk noise than for delay uncertainty, while simultaneously optimizing both delay and noise. In this section, we investigate a strategy, wherein, delay is considered as of higher criticality than noise, while simultaneously optimizing delay and noise. It is interesting to note that both methods yield significantly better optimization of delay and noise compared to other methods. The designer can choose either of the strategies based on the need. The difference between the two strategies is the way in which the sorted list is created. After the design is placed and routed, the path delays of all the paths in the design are estimated, and are sorted into a list based on their delay criticality. The most critical path in terms of delay is chosen to create games for gate size optimization. The games are created for each net in the chosen path in the order from its primary output to primary inputs. As an example for illustration, consider the chosen path to consist of four gates: A, B, C and D, in successive transition connected with nets: $1, 2$ and 3, respectively. The gate A is driven by primary inputs and gate D drives a primary output. In order to consider the down-stream load capacitance, the net 3 connecting gates C and D should be optimized before the nets 1 and 2. Thus, the games are played in the order of net 3 followed by net 2 followed by net 1. The game formulated for each net involves its driver, receiver and its two most aggressor gates. The two most aggressor gates for the net and its strategies are identified as indicated in section 7.3.1. After the games are played for all the nets of the chosen critical path in its direction of primary output to primary inputs, the next critical path in the sorted list is selected to play games. This process of creating games is repeated until the sorted list is empty.

7.4 Experimental Results

The stochastic game theoretic gate size solver (SGGS) described in Algorithm 7.1. was implemented in C and executed on a UltraSPARC-IIe 650MHz, 512MB Sun Blade 150 system running Solaris 2.8 on it. We have tested our gate sizing approach on several medium and large IP cores obtained from Opencores [62]. A standard cell library containing 10 logic cells with up to 8 different drive strengths based on a 6-Metal layer, 180nm technology has been developed and used. ASIC designs, written in behavioral VHDL/Verilog are converted to structural VHDL/Verilog using the standard cells in the library with the help of BuildGates, an RTL synthesis tool of Cadence design systems. We have modified the ASIC designs such that all the blocks in the design are flattened to standard cells in the library without maintaining the hierarchy. The on-chip memory modules are realized as D-flipflop register arrays. The structural VHDL/Verilog design is used as input by Cadence First Encounter to develop the floorplan. We have set the option of row utilization to 70% for all the designs so as to allow gate size scaling. The design is then placed and routed using Amoebaplace and Nanoroute respectively, which are part of Cadence First Encounter tool. The final placed and routed design is then exported in DEF format.

The parasitic information from the routed design is extracted in SPEF format using Synopsys StarRCXT. The interconnect resistance, interconnect capacitance, interconnect inductance, coupling capacitances along with their aggressor drivers are extracted from the SPEF file and is given as input to our SGGS. The length of overlap with the aggressor nets and their spacing is extracted from the DEF file and is also given as input. The calculated gate size for each gate is used to update the original DEF file to generate an optimized DEF file. It can be noted that the optimized DEF file is created with the help of gawk script and verified for DRC rules. The design is not re-routed to generate the optimized DEF file. The delay and crosstalk noise violations are measured using Cadence First Encounter's timing closure and CelticIC respectively with their robust models, and not using the analytical models used during optimization.

Table 7.1. Comparison of Stochastic Game Theoretic Approach with Deterministic and Geometric Programming Approach

Open Core Design [62]	Total Gates	Deterministic Approach			GP Approach [37]			SGT-NC Approach[1]			SGT-DC Approach[2]		
		Run Time (mins)	% Yield Improvement		Run Time (mins)	% Yield Improvement		Run Time (mins)	% Yield Improvement		Run Time (mins)	% Yield Improvement	
			Timing	Noise		Timing	Noise		Timing	Noise		Timing	Noise
PCI	7882	4.29	47.48	39.07	24.37	88.19	-	128.12	95.34	96.92	135.98	98.38	92.76
ATA	21781	10.49	53.94	42.47	73.52	90.41	-	376.04	89.75	94.72	391.68	95.29	91.46
RISC	34172	13.31	32.46	48.32	111.32	84.96	-	504.78	91.35	96.57	487.83	98.06	90.66
AVRμP	41274	15.79	46.18	48.02	135.49	93.36	-	621.29	97.56	98.04	633.15	97.81	95.69
P16C55	52128	19.98	28.95	31.45	171.55	83.59	-	783.10	93.75	97.11	768.85	98.72	95.86
T80μC	69973	27.67	30.27	22.74	278.92	79.21	-	947.54	95.08	98.69	928.61	98.17	96.33

* No area overhead for all four approaches. All the approaches are implemented at post layout level

[1] SGT-NC Approach: SGGS algorithm with gates ordered based on noise criticality for simultaneous minimization of delay uncertainty and crosstalk noise

[2] SGT-DC Approach: SGGS algorithm with gates ordered based on delay criticality for simultaneous minimization of delay uncertainty and crosstalk noise

Run Time: running time of each algorithm; *Yield*: fraction of test instances without timing or noise violations; *Timing Violations*: setup and hold time violations; *Noise Violations*: nets with noise values greater than $0.15V_{dd}$

In order to introduce the uncertainty due to process variations in benchmark circuits, we have modeled the probability distribution as the Gaussian $G(\mu, \sigma)$, with μ as the nominal gate size and 3σ as the 25% of the nominal gate size. We have generated 10,000 random samples for the gate sizes obtained from different gate sizing approaches using a Gaussian distribution for the process variations. This creates 10,000 instances of the same design with different gate sizes around its nominal values obtained by the gate sizing approaches. The timing yield of a design is measured as the fraction of test instances without set-up and hold time violations, while noise yield is measured as the fraction of test instances whose nets have an noise induced of exceeding a threshold value set as to $0.15V_{dd}$.

The existing works [34, 41, 42, 43, 35, 36, 37] on statistical gate sizing in the literature compare their results with a deterministic approach, but do not provide an comparison with other statistical gate sizing works. We have compared our results with a deterministic game theoretic gate sizing proposed in Chapter 5 and geometric programming based statistical gate sizing proposed in [37]. The gate sizing algorithm given in Chapter 5 is implemented as is and Gaussian distribution is used to create 10,000 instances of the design with gate sizes around the resulted nominal values generated by the algorithm. In [37] the authors have provided a geometric programming approach for tradeoff between area and robustness. In order to provide a fair comparison, we have implemented the geometric programming approach as illustrated in [37] for zero area overhead.

The experimental results are presented in Table 7.1.. Columns one and two give the name of the benchmark and the total number of gates in the design respectively. The columns three, six, nine and twelve give the run time of deterministic (see Chapter 5), geometric programming [37], stochastic games with games ordered according to noise (refer Section 7.3.1) and stochastic games with games ordered according to delay (refer Section 7.3.2) respectively. The columns four, seven, ten and thirteen give the timing yield results for the four approaches in their respective order, while columns five, eight, eleven and fourteen give the noise yield results respectively. It can be observed that the SGGS approach has better timing and noise yields when compared to geometric programming at the expense of increased run times.

7.5 Conclusions

Optimizing both delay uncertainty and crosstalk noise is extremely critical in deep submicron and nano regime circuits. The transition of states and the payoffs to players in each state of stochastic games are based on the probability distribution, which enables the designer to capture the variations in process parameters. Hence, the use of stochastic games essentially makes it possible for multi-agent optimization under the impact of process variations. The time complexity of finding the Nash equilibrium of each stochastic game is proportional to m^T, where m and T are parameters with spare sampling algorithm based solution and are not controlled by the circuit size. The values used for m and T provide a tradeoff between the quality of Nash solution and the run time. The designer can choose these values logistically which matches to his needs. The proposed stochastic game theoretic approach achieves better timing and noise yields when compared to deterministic and geometric programming.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

The current trends of device and interconnect scaling in CMOS circuits in continuing efforts to satisfy Moore's law have brought to light a numerous complex problems for both chip designers and manufacturers. With submicron device dimensions and more than hundred million transistors integrated on a single chip, the on-chip interconnects are playing a major role in determining the performance, the power consumption, the size and the reliability of digital systems. Some of the important and complex problems in deep submicron and nanometer designs are (i) increased interconnect delay due to rising RC parasitics of on-chip wiring, (ii) crosstalk noise due to increased coupling capacitance between adjacent wires (iii) increased power dissipation due to interconnects (iv) delay and noise uncertainty due to process variations. Several attempts have been made by the researchers to minimize the effects of one of the above concerns on DSM designs with others as the design constraints. But, it is interesting to note that the above mentioned problems are in conflict - minimizing the effects of one parameter will drastically deteriorate the effects of others parameters on the digital system. Hence, simultaneously optimizing two or more parameters of the digital systems becomes a challenging task. In this dissertation, the main focus is on simultaneously optimizing multiple parameters so as to collectively reduce their effects on VLSI circuits. The main contributions are in terms of developing new methods and algorithms for multi-metric optimization in deep submicron and nanometer designs.

In this dissertation, we have developed multi-metric optimization framework for performing gate sizing and wire sizing at post layout level. In specific, we have solved the following problems:

(i) A wire sizing framework for simultaneous optimization of interconnect delay and crosstalk noise - a two metric optimization framework

(ii) A wire sizing framework for simultaneous optimization of interconnect delay, power and crosstalk noise - a three metric optimization framework

(iii) A gate sizing framework for simultaneous optimization of interconnect delay and crosstalk noise - a two metric optimization framework

(iv) A gate and wire sizing framework for simultaneous optimization of interconnect delay and crosstalk noise - an integrated framework

(v) A statistical gate sizing framework for simultaneous optimization of delay uncertainty and crosstalk noise - a stochastic framework

(vi) New interconnect models based on transmission lines

The use of game theory models and Nash equilibrium to solve the post layout interconnect problems for simultaneous optimization of multiple parameters is unique to this dissertation. The development of a stochastic framework using stochastic games to solve the statistical gate sizing problem under the impact of process variations is novel. Game theoretic and auction models are previously used for power optimization at behavioral and logic levels of design abstraction [70]. However, the algorithms developed have an exponential time complexity. On contrary, in this dissertation, we have developed game theoretic models and algorithms for post layout multi-metric optimization with *linear time and space complexities*. The development of algorithms with linear complexities is another prime and novel contribution of this dissertation, especially considering todays DSM designs with a typical transistor count of hundred million or more. The publications resulted from this dissertation are given in references [71, 72, 73, 74].

Based on the results presented in this dissertation, we are encouraged to develop game theoretic and stochastic game models for multi-metric optimization framework with the following future directions:

(i) The game theoretic models developed in this dissertation can be used to solve the multi-metric optimization problem using gate sizing at logic level or RTL level.

(ii) Buffer insertion or repeater insertion is another technique which is widely used to minimize the interconnect effects on long interconnect wires. This technique would be effective if used at layout level while performing floorplanning or at logic level. A game theoretic framework can be developed to solve the buffer insertion problem for multi-metric optimization.

(iii) We have developed a stochastic framework for solving the statistical gate sizing problem considering the uncertainty in process parameters due to process variations. This can be extended to solve the statistical buffer insertion problem both during floorplanning and logic synthesis.

(iv) In Chapter 6, we have developed an integrated framework which can simultaneously perform both gate and wire sizing. In current DSM designs, there is a prime necessity for developing such integrated frameworks which can simultaneously apply two or more design techniques. Hence, a game theoretic framework which is capable of performing buffer insertion and gate sizing or buffer insertion and wire sizing or all three together can be a possibility after developing game theoretic algorithms for buffer insertion.

(v) In this dissertation, we have developed an objective function which can effectively handle two design parameters. We have also attempted to optimize three design parameters in Section 3.4. But, the objective function used was inefficient for handling three parameters, as seen from the experimental results. Hence, there is a need to formulate a new or improve the existing objective function which can effectively handle three or more design parameters.

(vi) In Chapter 7, the time complexity of the developed stochastic game based gate sizing algorithm is proportional to m^T, where m and T are constants. But, still this constant is exponential. The reason for this exponential constant is due to the use of *Sparse Sampling* algorithm to solve the Nash equilibrium of the stochastic games. To improve the time complexity of the proposed stochastic framework, new algorithms for solving the Nash equilibrium in 2-player stochastic games have to be developed.

REFERENCES

[1] ITRS. International technology roadmap for semiconductors. http://public.itrs.net, 2004.

[2] C. Hu. Future CMOS Scaling and Reliability. *Proc. of IEEE*, 81(5):682–689, May 1993.

[3] K. Rahmat, O. S. Nakagawa, S-Y. Oh, and J. Moll. A Scaling Scheme for Interconnect in Deep Submicron Processes. In *Technical Report HPL-95-77, Hewlett-Packard Laboratories*, pages 1–4, July 1995.

[4] V. Mehrotra, S. L. Sam, D. Boning, A. Chandrakasan, R. Vallishayee, and S. R. Nassif. A Methodology for Modeling the Effects of Systematic Within-Die Interconnect and Device Variation on Circuit Performance. In *Proc. of DAC*, pages 172 – 175, 2000.

[5] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De,. Parameter Variations and Impacts on Circuits and Microarchitecture. In *Proc. of DAC*, pages 338 – 342, 2003.

[6] J. Cong, L. He, K. Kei-Yong, K. Cheng-Ko, and D. Z. Pan. Interconnect Design for Deep Submicron ICs. In *Proc. of ICCAD*, pages 478–485, 1997.

[7] H. B. Bakoglu. *Circuits, Interconnections and Packaging for VLSI.* Addison-Wesley, 1990.

[8] K. T. Tang, and E. B Friedman. Interconnect Coupling Noise in CMOS VLSI Circuits. In *Proc. of ISPD*, pages 48–53, 1999.

[9] N. Shanbhag, K. Soumyanath, and S. Martin. Reliable Low-Power Design in the Presence of Deep Submicron Noise. In *Proc. of Intl. Symp. on Low Power Electronics and Design*, pages 295–302, 2000.

[10] M. R. Becer, D. Blaauw, I. Algor, R. Panda, C. Oh, V. Zolotov, and I. N. Hajj. Post-Route Gate Sizing for Crosstalk Noise Reduction. In *Proc. of ISQED*, pages 171–176, Mar. 2003.

[11] C.C. Chang, J. Cong, D. Zhigang, and X. Yuan. Interconnect-driven Floorplanning with Fast Global Wiring Planning and Optimization. In *Proc. of SRC Techcon Conf.*, pages 21–23, Sept. 2000.

[12] R. Sitte, S. Dimitrijev, and H. B. Harrison. Device Parameter Changes Caused by Manufacturing Fluctuations of Deep Submicron MOSFETs. *IEEE Trans. on Electron Devices*, 41(11):2210–2215, Nov. 1994.

[13] S. Nassif. Modeling and Forecasting of Manufacturing Variations. In *Proc. of ASP-DAC*, pages 145 – 149, 2001.

[14] X. Bai, C. Visweswariah, P. Strenski, and D. Hathaway. Uncertainty-Aware Circuit Optimization. In *Proc. of DAC*, 2002.

[15] A. Vetta. Nash Equilibria in Competitive Societies, with Applications to Facility Location, Traffic Routing and Auctions. In *Proc. of IEEE Symp. on Foundations of Comp. Sci.*, pages 416–425, Nov. 2002.

[16] F. Forgo, J. Szep, and F. Szidarovszky. *Non-Convex Optimization and its Applications: Introduction to the Theory of Games - Concepts, Methods, Applications*. Kluwer Academic Publishers, 1999.

[17] C. Papadimitriou. Algorithms, Games, and the Internet. In *Proc. of ACM Symp. on Theory of Computing*, pages 749–753, 2001.

[18] C. Papadimitriou. On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence. *Journal of Computer and System Science*, 48(3):498–532, 1994.

[19] J. Cong, and K. Cheng-Kok. Simultaneous Driver and Wire Sizing for Performance and Power Optimization. In *Proc. of ICCAD*, Nov. 1994.

[20] S. S. Sapatnekar. Wire Sizing as a Convex Optimization Problem: Exploring the Area-Delay Tradeoff. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 15(8):1001–1011, Aug. 1996.

[21] C. Chung-Ping, and D. F. Wong. Optimal Wire-Sizing Function with Fringing Capacitance Consideration. In *Proc. of DAC*, pages 604–607, Nov. 1997.

[22] J. Cong, and D. Z. Pan. Wire Width Planning for Interconnect Performance Optimization. *IEEE TCAD*, 21(3):319–329, Mar. 2002.

[23] C. C. N. Chu, and D. F. Wong. Closed Form Solution to Simultaneous Buffer Insertion/Sizing and Wire Sizing. *ACM TODAES*, 6(3):343–371, July 2001.

[24] J. I. H-Ru, C. Y-Wen, and J. J-Yang. Crosstalk-Driven Interconnect Optimization by Simultaneous Gate and Wire Sizing. *IEEE TCAD*, 19(9):999–1010, Sept. 2000.

[25] Jiang-An He, and H. Kobayashi. Simultaneous Wire Sizing and Wire Spacing in Post-Layout Performance Optimization. In *Proc. of ASP-DAC*, pages 373–378, 1998.

[26] T-C. Chen, S-R. Pan, and Y-W. Chang. Timing Modeling and Optimization Under the Transmission Line Model. *IEEE TVLSI Systems*, 12(1):28–41, Jan. 2004.

[27] T. Xiao and M. M. Sadowska. Gate Sizing to Eliminate Crosstalk Induced Timing Violation. In *Proc. of ICCAD*, pages 186–191, 2001.

[28] C. Alpert, C. Chu, G. Gandham, M. Hrkic, J. Hu, C. Kashyap, and S. Quay. Simultaneous Driver Sizing and Buffer Insertion Using a Delay Penalty Estimation Technique. In *Proc. of ISPD*, pages 104–109, 2002.

[29] C. Albrecht, A. B. Kahng, I. Mandoiu, and A. Zelikovsky. Floorplan Evaluation with Timing-Driven Global Wireplanning, Pin Assignment, and Buffer/Wire Sizing. In *Proc. of the Intl. Conf. on VLSI Design*, pages 580–587, 2002.

[30] M. Hashimoto, M. Takahashi, and H. Onodera. Crosstalk Noise Optimization by Post-Layout Transistor Sizing. In *Proc. of ISPD*, pages 126–130, 2002.

[31] D. Sinha, and H. Zhou. Gate Sizing for Crosstalk Reduction under Timing Constraints by Lagrangian Relaxation. In *Proc. of ICCAD*, pages 14–19, 2004.

[32] A. Murugavel and N. Ranganathan. Gate Sizing and Buffer Insertion using Economic Models for Power Optimization. In *Proc. Intl. Conf. on VLSI Design*, pages 195–200, 2004.

[33] E.T.A.F. Jacobs, and M.R.C.M Berkelaar. Gate sizing using a Statistical Delay Model. In *Proc. of the DATE*, pages 283–289, 2000.

[34] S. Raj, S. B. K. Vrudhula, and J. Wang. A Methodology to Improve Timing Yield in the Presence of Process Variations. In *Proc. of DAC*, pages 448–453, 2004.

[35] 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 *Proc. of ICCAD*, pages 1023–1028, 2005.

[36] D. Sinha, N. V. Shenoy, and H. Zhou. Statistical Gate Sizing for Timing Yield Optimization. In *Proc. of ICCAD*, pages 1037–1042, 2005.

[37] J. Singh, V. Nookala, Z-Q. Luo, S. Sapatnekar. Robust Gate Sizing by Geometric Programming. In *Proc. of DAC*, pages 315–320, 2005.

[38] A. Devgan, and S. Nassif. Power Variability and its Impact on Design. In *Proc. of Intl. Conf. on VLSI Design*, pages 679–682, 2005.

[39] A. Teene, B. Davis, R. Castagnetti, J. Brown, and S. Ramesh. Impact of Interconnect Process Variations on Memory Performance and Design. In *Proc. of ISQED*, pages 694–699, 2005.

[40] R. Rao, K. Agarwal, A. Devgan, K. Nowka, D. Sylvester, and R. Brown. Parametric Yield Analysis and Constrained-Based Supply Voltage Optimization. In *Proc. of ISQED*, 2005.

[41] A. Agarwal, K. Chopra, and D. Blaauw. Statistical Timing Based Optimization using Gate Sizing. In *Proc. of the DATE*, pages 400–405, 2005.

[42] O. Neiroukh, and X. Song. Improving the Process-Variation Tolerance of Digital Circuits Using Gate Sizing and Statistical Techniques. In *Proc. of the DATE*, 2005.

[43] M. R. Guthaus, N. Venkateswaran, C. Visweswariah, and V. Zolotov. Gate Sizing Using Incremental Parameterized Statistical Timing Analysis. In *Proc. of ICCAD*, pages 1029–1036, 2005.

[44] D. Velenis, R. Sundaresha, and E. Friedman. Buffer Sizing for Delay Uncertainty Induced by Process Variations. In *Proc. of Intl. Conf. on Electronics, Circuits and Systems*, pages 415–418, 2004.

[45] J. Xiong, K. Tam, and L. He. Buffer Insertion Considering Process Variation. In *Proc. of the DATE*, 2005.

[46] E. Rasmusen. *Games and Information: An Introduction to Game Theory.* Blackwell Publishers, 3rd edition, 2001.

[47] M. Bardi, T. E. S. Raghavan, and T. Parthasarathy. *Stochastic and Differential Games: Theory and Numerical Methods.* Birkhauser Boston Publishers, 1999.

[48] A. P. Maitra, and W. D. Sudderth. *Discrete Gambling and Stochastic Games.* Springer-Verlag Publishers, 1996.

[49] J. von Neumann. Zur Theorie der Gesellschaftsspiele. *Zur Theorie der Gesellschaftsspiele*, pages 295–320, 1928.

[50] J. F. Nash. The bargaining problem. *Econometrica*, 18(2):155–162, Apr. 1950.

[51] T. Roughgarden. Stackelberg Scheduling Strategies. In *Proc. of the ACM*, pages 104–113, 2001.

[52] J. F. Nash. Equilibrium Points in N-Person Games. *Proceedings of the National Academy of Science of the United States of America*, 36(1):48–49, Jan. 1950.

[53] G. Owen. *Game Theory.* Academic Press, UK, 1995.

[54] Marcin Jurdzinski Krishnendu Chatterjee and Rupak Majumdar. On nash equilibria in stochastic games. Technical Report UCB/CSD-03-1281, EECS Department, University of California, Berkeley, 2003.

[55] S. Seki, and H. Hasegawa. Analysis of Crosstalk in Very High-Speed LSI/VLSIs Using a Coupled Multi-Conductor Stripline Model. *IEEE Trans. Microwave Theory Tech.*, MTT-32:1715–1720, Dec. 1984.

[56] T. Sakurai, and K. Tamaru. Simple Formulas for Two- and Three-Dimensional Capacitances. *IEEE Trans. Electron Devices*, ED-30:183–185, Feb. 1983.

[57] C. S. Walker. *Capacitance, Inductance, and Crosstalk Analysis.* Artech House, Boston, 1990.

[58] D. Sylvester, and C. Hu. Analytical Modeling and Characterization of Deep Submicrometer Interconnect. *Proc. of the IEEE*, 89(5):634–664, May 2001.

[59] C. J. Alpert, A. Devgan, and S. T. Quay. Is wire tapering worthwhile? In *Proc. of ICCAD*, pages 430–435, 1999.

[60] S. P. Khatri, R. K. Brayton, A. L. Sangiovanni-Vincentelli. *Cross-Talk Noise Immune VLSI Design using Regular Layout Fabrics.* Kluwer Academic Publishers, 2001.

[61] S. Kakutani. A Generalization of Brouwer's Fixed Point Theorem. *Duke Journal of Mathematics*, 8:457–459, 1941.

[62] Open Cores. Free Open Source IP Cores and Chip Design. http://www.opencores.org.

[63] CRETE. Cadence Repository for Electronic Technical Education. http://crete.cadence.com.

[64] GALib. GAlib - A C++ Library of Genetic Algorithm Components. http://lancet.mit.edu/ga/.

[65] L. N. Dworsky. *Modern Transmission Line Theory and Applications.* John Wiley and Sons.

[66] LANCELOT. A Package for Large-Scale Nonlinear Optimization. http://www.numerical.rl.ac.uk/lancelot/blurb.html.

[67] H. Hashimoto, and H. Onodera. Increase in delay uncertainty by performance optimization. In *Proc. of ISCAS*, pages 379–382, 2001.

[68] P. Ghanta, S. Vrudhula, R. Panda, and J. Wang. Stochastic Power Grid Analysis Considering Process Variations. In *Proc. of the DATE*, 2005.

[69] M. Kearns, Y. Mansour, and S. Singh. Fast Planning in Stochastic Games. In *Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence*, pages 309–316, 2000.

[70] A. Murugavel. New methods for dynamic power estimation and optimization in VLSI circuits. Ph.D. Thesis, USF LIBRARY, Tampa Circulating Collection - TK7885.Z9 M87 2003, University of South Florida, Tampa, FL, 2003.

[71] N. Hanchate and N. Ranganathan. Simultaneous Interconnect Delay and Crosstalk Noise Optimization through Gate Sizing Using Game Theory. *IEEE Trans. on Computers (To Appear)*, 2006.

[72] N. Hanchate and N. Ranganathan. A Game-Theoretic Framework for Multi-Metric Optimization of Interconnect Delay, Power and Crosstalk Noise During Wire Sizing. *ACM TODAES (To Appear)*, 2006.

[73] N. Hanchate and N. Ranganathan. A Linear Time Algorithm for Wire Sizing with Simultaneous Optimization of Interconnect Delay and Crosstalk Noise. In *Proc. of Intl. Conf. on VLSI Design*, pages 283–290, 2006.

[74] N. Hanchate and N. Ranganathan. Post-Layout Gate Sizing for Interconnect Delay and Crosstalk Noise Optimization. In *Proc. of ISQED*, 2006.

ABOUT THE AUTHOR

Narender Hanchate received the B.E. degree in Electronics and Communication Engineering from Osmania University, Hyderabad, India, in 2000, and the M.S. degree in Computer Science and Engineering from the University of South Florida, Tampa, in 2003. He is currently working towards his Ph.D. degree in Computer Science and Engineering at University of South Florida, Tampa, FL. He is a student member of IEEE and IEEE Computer Society. From 2000 to 2001, he worked as a Design Engineer in the field of VLSI and Embedded Systems at HCL Technologies Ltd, Chennai, India. His research interests include the development of methodologies for low-power synthesis, multi-metric optimization, development of methodologies and tools for interconnect optimization, and process variations in deep submicron designs.