A Study on Performance Optimization for Digital CMOS Circuits in Physical Design

Masanori Hashimoto

Abstract

This thesis discusses circuit optimization for performance enhancement in physical design. The target of the performance optimization methods discussed in this thesis is digital CMOS circuits. Due to steady improvement in LSI fabrication technology, LSI designers encounter various problems that are not critical so far. This thesis focuses on the following problems and proposes some solutions in physical design for each problem; power dissipation, delay fluctuation, and crosstalk noise problems.

In this research, a delay and power optimization method by input reordering is developed. This method utilizes the characteristics difference of delay and power between logically-equivalent input pins for performance enhancement. The effectiveness is experimentally verified using 30 benchmark circuits. This method reduces power dissipation by 22.5% maximum and by 5.9% on average. Delay time is also reduced by 6.7%. A gate sizing method that reduces glitches for power reduction is devised. This method optimizes power dissipation with a statistical glitch estimation method and an efficient gate sizing algorithm. The proposed method is experimentally examined using 10 circuits. Power dissipation is reduced by 16.2% maximum and by 10.4% on average further from minimum-area circuits that are regarded as minimum-power circuits by conventional methods.

Next, a statistical timing analysis method that can handle local random delay fluctuation is improved in accuracy. Also a delay and power optimization method by gate sizing based on statistical timing analysis is developed. A new measure that represents timing criticality at each cell is devised, which improves the efficiency of optimization algorithm. The proposed method contributes to exclude over-design and under-design of LSI. This thesis also demonstrates some examples that performance optimization increases delay uncertainty, and verifies that the proposed statistical timing analysis method is effective as one of solutions for this problem.

This thesis proposes a design methodology that transistor sizes are continuously varied inside cells while keeping cell-base design framework. This design methodology aims to design a high-performance circuit whose performance is close to that of full custom design. Exploiting this design methodology, a power reduction method that downsizes transistors inside cells after detail-routing is developed. The effectiveness is experimentally verified using 5 circuits. The proposed method reduces power dissipation by 65% on average without delay increase compared with usual cell-based circuits. The proposed design methodology can vary transistor sizes after detail-routing in spite of preserving interconnects. A crosstalk reduction method by transistor sizing, which utilizes this feature thoroughly, is developed.

This method estimates crosstalk noise inside optimization loops using the interconnect information extracted from detail-routed layout. Finally the layout, which the optimization result is applied to, is obtained without any interconnect modifications. The experimental results in 2 circuits show that the maximum noise voltage is reduced by more than 35% without any delay increase.

Acknowledgments

I would like to express my gratitude to Professor Hidetoshi Onodera in Kyoto University for his smart guidance and robust leadership throughout this research. I would like to thank Professor Yukihiro Nakamura, Professor Toru Sato, and Professor Yahiko Kambayashi for their profitable advice on writing this thesis. I am grateful for technical suggestions with Honorary Professor Keikichi Tamaru and Professor Kazutoshi Kobayashi in Kyoto University. The work in Chapter 7 is assisted by Mr. Masao Takahashi in Onodera Laboratory. I am thankful for his cooperation. This work is supported in part by Semiconductor Technology Academic Research Center (STARC). I would like to show my appreciation of useful advice from the co-researchers; Dr. Nakayama of STARC, Dr. Ogawa of Hitachi, Dr. Takeuchi of Fujitsu, and Dr. Nishimoto of Sharp. I would like to express my thanks to the members in Onodera Laboratory who contribute to this work through active discussions; Dr. Akio Hirata, Mr. Ryota Nishikawa, Mr. Tetsutaro Hashimoto, Mr. Daisuke Fukuda, and Mr. Kazunori Fujimori. I appreciate the financial support from the Japan Society for the Promotion of Science. I am heavily indebted to my wife Kimiko through daily life. I am also under an obligation to my mother Kyoko. Finally, I hope my gratitude will reach to my father Yoshiaki in heaven.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter discusses the research motivations and the contributions of this thesis. First the problems that LSI designers encounter in physical design phase are discussed. The trends of research for each problem are also discussed. These problems are expected to become more serious as fabrication technology advances, because these problems are originated in shrinking feature size. Then, the objective of this research and the organization of this thesis are explained.

1.1 Problems in Physical Design Due to Deep Submicron Technology

Due to severe competitions between LSI(Large Scale Integrated Circuit) design companies, the design of high-performance and high-functionality LSIs are requested to LSI designers. Although the number of devices that can be integrated on a single chip increases exponentially as the process technology improves, the design-time assigned to an LSI design gets shorter, since the life-time of new products becomes short. Therefore, the demand for design automation keeps on rising, and design automation has been hoped to cover larger area of LSI design. So far, the design phase called "physical design" is one of the most advanced area in design automation, and the most part of physical design can be automated using CAD(Computer Aided Design) tools. Here, in physical design, the layout of devices is generated and placed devices are connected by wiring. Also, the circuit is partially modified to satisfy performance requirements. With the invention of automatic placement and routing tools and logic synthesis tools, which construct a gate-level netlist from RTL(Resister Transfer Level) descriptions, a chip that contains more than million gates can be designed.

With the advent of the deep submicron era, a set of issues that circuit designers are faced with is vastly different from those in traditional designs. Needless to say, some serious problems caused by DSM(Deep SubMicron) process emerge in physical design as well as in other design phases. Especially, the following issues are considered as severe problems; 1) interconnect delay, 2) power dissipation, 3) delay fluctuation and 4) crosstalk noise. Hereafter, each problem is discussed.

1.1.1 Interconnect Delay

Interconnects become one of the dominant factors that determine the circuit performance. One of the reasons that interconnects limit performance is wiring capacitance. With increasing chip dimensions, interconnect capacitance dominates gate capacitance, and the speed improvement expected from simple scaling does not apply to circuits that drive global wires. Simple scaling assumes a reduction in capacitive loading due to wires. This is true locally when a circuit is connected only to its neighbors, but for circuits that drive long global wires, the capacitive loading actually increases because chip size gets larger with shrinking. This delay increase caused by capacitive load of interconnect can be reduced by adjusting the driver strength. Many gate/transistor sizing methods that optimize the size of gate/transistor have been proposed [1, 2, 3, 4, 5, 6, 7]. Buffer insertion methods, which divide a heavy load into smaller loads and/or isolate a heavy load from critical path, are also studied [6, 7, 8, 9].

In addition to large capacitance loads resulting from long interconnects, the resistance of the lines also becomes a major concern. The most commonly cited DSM interconnect problem is that of rising RC wire delays. It can be clearly seen that wiring delay is capable of consuming the majority of the shrinking clock cycle time in DSM designs. The 50%-to-50% delay which includes both gate and interconnect delay is expressed as follows[10].

$$T_{50\%} = 0.377 R_{int} C_{int} + 0.693(R_{tr} C_{int} + R_{tr} C_L + R_{int} C_L), \quad (1.1)$$

where, R_{int}, C_{int} is the total resistance and capacitance of the interconnect. R_{tr} is the output resistance of the driver, and C_L is the load capacitance connected to the end of the interconnect. The first term of $0.377 R_{int} C_{int}$, which corresponds to the distributed RC delay, becomes dominant as the interconnect length increases, since the value of $R_{int} C_{int}$ is proportional to the square of the interconnect length. Table 1.1 shows the trend of interconnect predicted in Ref. [11], and Table 1.2 lists the values of resistance, capacitance, and RC product of interconnect in future technology. RC delay is increasing as the technology advances, though the low-resistive metal is used for interconnect and the low-dielectric insulators are developed. The distributed RC delay of interconnect cannot be reduced by increasing the driving strength, because it is independent of drier size. Dividing the interconnect into some segments by inserting repeaters is the most effective solution, and many techniques have been proposed [12, 13, 14, 15, 16]. Wire sizing is also effective to reduce wiring RC delay and has been researched[17, 18, 19, 20, 21, 22, 23].

As described above, the delay time caused by interconnect capacitance and resistance occupy a large amount of the total circuit delay in DSM technology. Traditionally, the phase of the logic design which utilizes logic synthesis tools is followed by layout design. In this phase, placement of cells and routes of interconnects are not fixed. The wire capacitance is statistically modeled according to the database that storages past designed circuits. The timing design is executed based on this statistical model of wire capacitance. In DSM processes, the capacitance difference between the real interconnects after routing and virtually-assumed interconnects becomes a critical problem in timing design. The number of iterations between logic synthesis and physical design increases, and the timing convergence becomes difficult. In order to solve this timing closure problem, some CAD vendors have proposed the methods that combine logic synthesis and placement. These methods incrementally modify the circuit structure based on the cell placement, as well as adjusting the driving strength of cells and inserting buffers, which advances the timing closure. Thus, much energy and effort of many researchers have been concentrated on the interconnect delay problems, and hence the solutions for reducing interconnect delay have been intensively explored.

Table 1.1: SoC Interconnect Technology Requirements[11].

Year	1999	2002	2005	2008	2014
Gate Length[μm]	0.18	0.13	0.10	0.07	0.035
Local(Cu): AR	1.4	1.5	1.7	1.9	2.2
Intermediate(Cu): AR	2.0	2.2	2.4	2.5	2.9
Global(Cu): AR	2.2	2.5	2.7	2.8	3.0
Effective Dielectric Constant(κ)	4.0	3.5	2.2	1.5	<1.5

AR is a wiring aspect ratio defined as height/width.

Table 1.2: Resistance(Cu) and Capacitance of Interconnects.

Process Technology[μm]		0.18	0.13	0.10	0.07	0.035
Local	Resistance[Ω/μm]	0.251	0.440	0.737	1.353	4.432
	Capacitance[fF/μm]	0.161	0.139	0.090	0.062	0.064
	RC product(1mm)[ns]	0.040	0.061	0.066	0.084	0.284
Intermediate	Resistance[Ω/μm]	0.107	0.185	0.317	0.611	2.294
	Capacitance[fF/μm]	0.197	0.173	0.110	0.074	0.076
	RC product(3mm)[ns]	0.190	0.288	0.314	0.407	1.569
Global	Resistance[Ω/μm]	0.036	0.060	0.104	0.207	0.813
	Capacitance[fF/μm]	0.219	0.195	0.121	0.081	0.080
	RC product(5mm)[ns]	0.197	0.293	0.315	0.419	1.626

1.1.2 Power Dissipation

Recently, reducing power dissipation has become a major concern in LSI design. In CMOS circuits, most of energy is consumed by charging and discharging capacitance, and hence power dissipation is represented as follows.

$$Power = \frac{1}{2} \cdot f \cdot V_{DD}^2 \sum_i C_i \cdot P_{sw}(i), \quad (1.2)$$

where f is the operating frequency and V_{DD} is the supply voltage. C_i is the capacitance of the i-th node and $P_{sw}(i)$ is the switching probability of the i-th node. When all nodes are assumed to have the same value of $P_{sw}(i)$ for simplicity, power dissipation is proportional to

the total capacitance in a circuit and the operating frequency. Due to the decrease in feature size, the operating frequency of the circuit increases. The total capacitance in a chip also increases as the number of integrated devices and the die size become large. Consequently, the latest chips tend to consume more power dissipation even though supply voltage decreases.

One of the major reasons why low power design is required is the increase in portable electronic equipments, such as laptop computers, cellular phones, and portable audio players. The power of these potable products are generally supplied from batteries, which limits the power dissipation of chips and encourages low power LSI design. High power dissipation involves overheating chips, which degrades performance and reliability and reduces chip life-time. In order to control the temperature, high power chips require costly specialized packaging and heat-sink arrangements. High power dissipation means high current density in a circuit. The supply voltage in a circuit is reduced by resistive voltage drops, which degrades the performance and may cause a failure. The extensive current density in wires also causes electromigration problems. The metal atoms migrates because of the collision of metal atoms and electrons, which results in electrical opens and shorts. Therefore, the estimation and reduction methods of power dissipation are strongly demanded in order to design a high-performance, high-competitive, high-reliable and low-cost chip.

1.1.3 Delay Fluctuation

The maximum operating speed is different in chip by chip, even when chips are fabricated using the same mask patterns, which is widely recognized as manufacturing variability. The circuit speed of fabricated chips is also different from the speed expected by circuit designers. It is because there are several sources that give rise to uncertainties in circuit delay. The sources that cause delay uncertainty can be categorized into two groups. The first group is physical fluctuation which is caused by the variabilities of physical parameters, such as length and width of MOSFETs, electrical characteristics of MOSFETs, shapes of interconnects, supply voltage and temperature, and so on. In the fabrication process of LSI, fabricating conditions necessarily fluctuates. This manufacturing variability varies the size and characteristics of devices, which results in delay fluctuation. The delay of each cell depends on supply voltage and temperature, and hence the change of operation condition is also a source of delay uncertainty. The second group of the uncertainty sources is design uncertainty. The design uncertainty contains the errors of cell delay model and RC extraction. It also includes noise, such as IR-drop, crosstalk, and etc. The problem is that the uncertainty sources can not be eliminated completely even though the amount of fluctuation may be reduced by various ways. Therefore, the design methodology that can consider the delay uncertainty is necessary to design high-performance and high-yield chips.

From the appearance of fluctuation, the fluctuation can be classified into two categories. The first category is a global change that applies to all gates and wires similarly in a certain region. The second category is a random change that indicates a certain statistical distribution. As for the global change, the worst-case analysis method is widely-used. The best/typical/worst-case delay times are calculated for each gate and wire, and then the circuit delay time is evaluated using a suitable case value for purpose. This is a reasonable approach

for the global change.

So far, the random change is scarcely considered, because the global change occupies a large mount of delay fluctuation. When the local fluctuation is small compared with the global change, setting a little design margin is sufficient to consider the local delay uncertainties. As the feature size becomes small, however, the effect of the local change becomes strong, and the local change can not be neglected now. References [24, 25] report that the local delay fluctuation caused by manufacturing variability is comparable with the global delay change in DSM technology. In the case that the design margin is set to large, it is sure that the under-design of circuits can be avoided. However the circuit is over-designed, i.e. the chip area and the power dissipation become large wastefully. Furthermore, as the performance requested for LSI design becomes high, there become many cases that the circuit performance can not be satisfied because of the settled over-margin. Therefore, in order to design a circuit with high performance and eliminate over-design, delay evaluation and optimization methods that can consider the random change are necessary.

1.1.4 Crosstalk Noise

Increasing interconnect resistance is the main reason for the increased wiring RC delay in DSM technology. Resistance is inversely proportional to the cross-sectional area of the wire. Due to the rising need for higher densities on chip, wiring pitches are decreased rapidly at about the same rate as gate length. In order to prevent resistance from increasing too quickly, line thickness(or height) is scaled at a slower rate, which results in taller, thinner wires. For example, Ref. [11] predicts an increase in wiring aspect ratio(AR=height/width) of local wires from 1.4 at a 0.18μm process to 2.2 at a 0.07μm process(Table 1.1). These lines with high aspect ratio involve an undesirable secondary effect that a large amount of coupling capacitance is brought out. In addition, spacing between wires is shrinking quickly in order to maintain high packing densities, coupling capacitance is further increased. It is reported that line-to-line capacitance between wires on the same level can be seen to make up over 70% of the total wiring capacitance at lower levels even at 0.25μm technologies[26].

Because of coupling capacitances, two signals at adjacent wires are affected each other. When a signal transition occurs at the neighboring wire, the transition propagates through the coupling capacitance, and a noise appears at the corresponding wire. This noise, which is called crosstalk noise, has become a critical problem in DSM LSI design. The problem caused by crosstalk noise is classified into two categories; dynamic delay variation and deterioration in signal integrity. The dynamic delay variation depends on the relative timing of the transitions occurred at neighboring wires. When the transition timings are close enough, the delay time of each transition are varied. The direction of the transitions is also an important factor. When both transitions are in the same direction, each delay time becomes short. Conversely, the directions of the transitions are different, the delay time increases. The deterioration in signal integrity may cause a functional failure. When the swing voltage of noise becomes larger than the logical threshold voltage, the logical value of the output gate changes, which is a serious problem especially in dynamic circuits. In order to avoid these timing and functional failures, the estimation and reduction methods of crosstalk noise

are necessary for DSM LSI design.

1.2 Overview of This Thesis

As explained in Section 1.1, there are several severe problems in physical design field. The major four problems are explained; interconnect delay, power dissipation, delay fluctuation and crosstalk noise. These problems depend on both circuit and layout. In DSM technology, a part of logic/circuit design has to be merged into physical design, that is to say, circuit optimization techniques need to consider placement and wiring in order to design high-performance circuits. All of performance metrics, such as circuit delay, power dissipation, and circuit area, have to be optimized considering delay fluctuation and crosstalk noise problems in order to ensure correct behavior of circuits.

The first problem of interconnect delay is intensively studied, and many techniques to control and reduce the interconnect delay, such as gate/transistor sizing, buffer insertion and wire sizing, have been proposed. Compared with interconnect delay, the rest of problems have not been explored thoroughly. The conventional circuit optimization methods leave space for consideration in power dissipation. The design methodologies that considers local delay fluctuation sufficiently have not been established. The estimation methods of crosstalk noise are now widely studied, but the effective circuit optimization methods to reduce crosstalk noise have not been proposed. The aims of this thesis are investigating the problems of power dissipation, delay fluctuation and crosstalk noise, and developing solutions for each problem. These problems are expected to become more serious, as fabrication technology improves. The methods proposed in this thesis are expected to become essential in future. This research contributes to design high-performance and high-reliability LSIs.

Thanks to the steady improvement of fabricating technology, SoC(System On a Chip) is turning into reality, i.e. a whole system can be integrated on a single chip. ASICs(Application Specific Integrated Circuit) that include System LSI are usually designed using a standard cell library. Generally, foundries or library vendors design and characterize standard cells beforehand, and provide a set of them as a standard cell library for each fabrication technology. The provided standard cell library is commonly used for designing chips fabricated in the same fabrication technology. The detailed characteristics of each cell, such as delay time and power dissipation, can be easily obtained due to exhaustive pre-characterization results, which make it easy to analyze the circuit performance and the circuit behavior. The design using a standard cell library is hence suitable for design automation, and is widely adopted. The framework of cell-base design for ASICs is well constructed. In cell-base design, circuit optimization during physical design is executed by replacing, inserting, and removing cells.

Cell-base design, however, limits the extent of design freedom for the benefit of the design facility. The flexibility of transistor sizes is highly restricted, since the circuit has to be composed by the pre-designed cells. Consequently, cell-based circuits make a sacrifice of optimality, and contain redundancy, for example, in power dissipation. In order to reduce this redundancy and get the high-quality circuits close to those of full-custom design, a cell-

layout generation system that can generate various driving-strength cells has been developed in our research group[27]. This system, which is called VARDS, can vary each transistor width inside a cell easily and flexibly. Here, the cells, which are generated on the fly according to the demand, are called "ondemand cells". This property enables transistor-level circuit optimization in the following design flow while keeping the cell-base design framework. At first, a cell library that includes several driving-strength variations for each logic, for example, x1, x2, x4, x8, where this composition is the same with the conventional cell library. Using this cell library, logic synthesis and cell placement are executed as usual. Before and/or after routing, the circuit is optimized in transistor-level. According to the optimization result, the cell layouts are generated on the fly, and each cell is replaced by the corresponding ondemand cell. The ondemand cell layouts have the same structure, i.e. the cell height is the same and the locations of power and ground metal are the same, which enables the layout design using a usual placement and routing tool. Thus, transistor-level circuit design can be realized, making the best use of cell-base design framework. Recently the parameterized standard-cell library, which is called "p-cell" library, is proposed[28]. The cells are parameterized by a continuous metric, gain. The logic synthesis is executed according to the gain information of each logic cell. Each cell layout is generated from the gain parameter, and the initial layout is constructed by those cells. Thereafter the layout is optimized by transistor-level circuit optimization techniques used in full-custom design methodology. This approach is based on full-custom design methodology for high-end chips, such as microprocessors used in mainframe computers, and aims to introduce a part of cell-base methodology for design efficiency. Therefore the logic synthesis tool and the layout tool are different from the tools used in usual cell-base design framework. On the other hand, the target of the "ondemand cell" approach is SoC and ASICs. The proposed methodology is extending cell-base design to full-custom design in part with the minimum effort, maintaining the cell-base design framework.

The circuit optimization methods discussed in this thesis optimize a block/module in LSIs. Reference [26] reports that RC distributed delay does not become dominant inside a block whose circuit scale is below 50k gates. Therefore, tuning driving strength, i.e. gate sizing and transistor sizing, is most effective and essential for the high-performance block design. The other methods to enhance the performance of blocks are buffer insertion and input reordering. The performance optimization by gate/transistor sizing and input reordering is studied in this thesis, and circuit optimization techniques for solving the problems discussed in Section 1.1 are discussed. This thesis focuses on gate/transistor sizing, and proposes two optimization method for low power, and a performance optimization method that can handle delay fluctuation. Also a power and delay optimization method by input reordering is discussed. In addition, this thesis proposes a transistor sizing method for crosstalk noise reduction. Conventionally interconnect optimization is widely executed. However, circuit optimization is hardly utilized for noise reduction, because circuit optimization involves interconnect modifications and interconnect modifications may spoil optimization results. The proposed method optimizes detail-routed circuits without any modifications thanks to "ondemand cell", and hence reduces crosstalk noise efficiently by circuit optimization.

This thesis is organized as follows. In Chapter 2, power and delay optimization by in-

put reordering is discussed. Due to the cell structure, the power and delay characteristics among the input pins in a cell are different even though the logical function is the same. This method utilizes the characteristics difference to improve performance by reordering the input pins. In Chapter 3, a power optimization method by gate sizing is discussed. This method considers that glitch transitions heavily depend on delay characteristics, and this sensitivity of glitches is also utilized for power reduction. The proposed method reduces the number of glitches as well as the amount of capacitive load and short-circuit current, whereas the conventional methods assume the number of glitches to be constant. Chapter 4 discusses a performance optimization method based on a statistical static timing analysis. This method focuses on the local fluctuation component of delay uncertainties, and calculates the statistically-distributed circuit delay. The aims of this method are the realization of high-performance and high-reliability LSI design and the removal of over-design and under-design of LSI. The performance optimization methods discussed in Chapters 2, 3 and 4 can be applied to both usual cell-base design and ondemand cell-base design.

In Chapter 5, a post-layout transistor sizing method is discussed. This method exploits ondemand-cell generation system to reduce power dissipation, and realizes high-performance circuit design close to full custom design. This method can optimize the detail-routed circuits without any modifications of interconnects, thanks to the feature of VARDS that the location of input and output pins are fixed while the transistor widths inside a cell are varied[27]. In Chapter 6, the delay uncertainty in the circuits optimized for performance enhancement is examined in a statistical way. Performance optimization has an aspect of path-balancing operation, i.e. the delay times of many paths are equalized. Due to the statistical characteristics, the optimized circuit becomes more sensitive to delay uncertainty, which results in the increase in circuit delay. Some examples of this problem are demonstrated, and then the statistical static timing analysis method discussed in Chapter 4 is evaluated as one of solutions. The discussion in Chapter 6 applies to not only performance optimization in physical design but also performance optimization in other design phases. Chapter 7 discusses a transistor sizing method that reduces crosstalk noise in detail-routed circuits. Crosstalk noise is heavily depends on the interconnect structure, so crosstalk noise can not be estimated until detail-routing completes. The circuit optimization for crosstalk noise reduction can be hardly executed after detailed-routing. This is because the wiring is also changed by circuit optimization, which may increase the crosstalk noise, or cause a new crosstalk noise problem. However, in the "ondemand cell" methodology, the transistor sizes inside cells can be optimized preserving interconnects as explained in Chapter 5. Crosstalk noise depends on the driving-strength of aggressive wire strongly, and hence crosstalk noise can be reduced by down-sizing transistors that drive the aggressive wires, with the information of adjacent wires extracted from the detail-routed layout. The methods discussed in Chapters 5 and 7 exploit the feature of ondemand cells. Finally Chapter 8 concludes this thesis.

Chapter 2

Performance Optimization by Input Reordering

This chapter discusses a method for power and delay optimization by input reordering. It is observed that the reordering has a significant effect on the power dissipation of the gate which drives the reordered gate. This is because the input capacitance depends on the signal values of other inputs. This property, however, has not been utilized for power reduction. Previous approaches focus on the reduction of the power dissipated by internal capacitances of the reordered gate. A heuristic algorithm considering the total power consumed in the driving gate and the reordered gate is devised. Experimental results using 30 benchmark circuits show that the proposed method reduces the power dissipation in all the circuits by 5.9% on average. There is a possibility that power dissipation is reduced by 22.5% maximum. In the case of delay and power optimization, the proposed method reduces delay by 6.7% and power dissipation by 5.3% on average.

2.1 Introduction

In the various stages of the VLSI design, many techniques for power reduction have been proposed, such as supply-voltage scaling[29, 30], technology mapping for low power[31], gate sizing[32], input reordering[33, 34, 35, 36, 37], and so on. The technique of input reordering has two advantages. The first advantage is that input reordering has little effect on the layout area. The second is that other techniques can be combined easily with input reordering. In [33], the authors proposed that an input with high switching probability should be connected with a pin which has small input capacitance. Here, a small input capacitance means that the size of the input transistor is small. However, this strategy is not effective in cell-base design because the pins that are equivalent logically have the same transistor size in most standard cell libraries. In [34, 35, 36, 37], the authors discussed input reordering for power reduction such that the reordering reduces the power dissipation inside the reordered gate. The input reordering, however, affects not only the power dissipated inside the reordered gate but also the power dissipated by a fan-in gate and fan-out gates, where the fan-in gate

is the gate which drives the reordered gate and the fan-out gates are the gates driven by the reordered gate. The effect of input reordering appeared in the fan-in gate comes from the fact that the input capacitance of the reordered gate differs depending on the signal values of other inputs, as explained in detail later. As a result, dynamic power dissipation of the fan-in gates changes according to the input reordering of the reordered gate. The variation of input capacitances has not been utilized for power reduction previously. This chapter discusses the effects of input reordering on power dissipation in the fan-in gate and the fan-out gates as well as in the reordered gate, and an improved method for power optimization which exploits the effects is proposed.

This chapter is organized as follows. Section 2.2 discusses the effects of input reordering on power dissipation and delay. Section 2.3 discusses strategies of input reordering for power and delay optimization at each gate. Section 2.4 introduces an algorithm of input reordering for power and delay optimization for the whole circuit. Section 2.5 shows the experimental results of the proposed method. Finally Section 2.6 concludes the discussion.

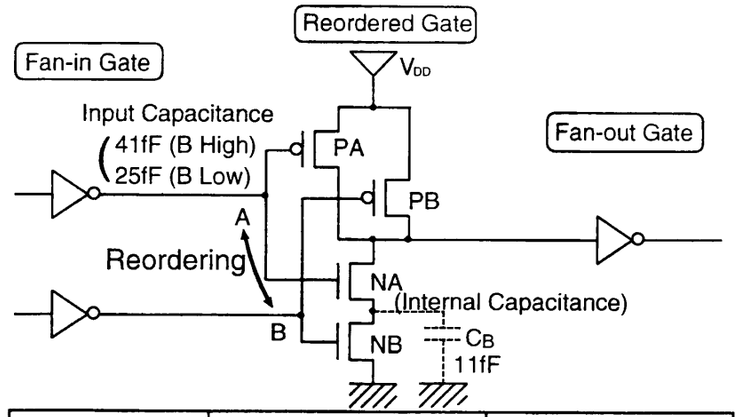
2.2 The Effects on Power Dissipation and Delay

This section discusses the major effects of input reordering in the fan-in gate, the reordered gate and the fan-out gate. So far, only the effect for the reordered gate has been considered for performance optimization. It is shown that there is a notable effect of input reordering on power dissipation in the fan-in gate, which could be utilized for performance optimization as well.

2.2.1 Fan-in Gate

The dynamic power dissipated by a fan-in gate varies by the input reordering of the reordered gate(the gate that the fan-in gate drives). This is because the input capacitance of the reordered gate, i.e. the load capacitance of the fan-in gate, depends on the signal values of other inputs of the reordered gate. The difference is demonstrated numerically using an example from a real 0.7 μm standard cell library. Fig. 2.1 shows a 2-input NAND gate with inputs A and B, two nMOSFETs NA and NB in series, being NA closer to the output. When the input B keeps low, the input capacitance of A is 25 fF. When the input B keeps high, the input capacitance of A becomes 41 fF which is 64% larger than the previous case. The difference of the input capacitance (16 fF) is larger than the internal capacitance(C_B = 11 fF) which is the sum of the diffusion capacitances of the source(NA) and the drain(NB).

The input capacitance of A depends on whether the source of NA is connecting to ground or not. Let us show that the input capacitance is small when the source of the input transistor is floating from ground, using a 3-input NAND gate (Fig. 2.2) as an example. Fig. 2.2 shows the method of measuring the input capacitance. A current meter i is added to measure the current poured into the input capacitance of B. The voltage source V_{in} generates a ramp waveform changing from 0 to V_{DD}. The integration of the current yields the charge Q_B



Fan-in Gate	Reordered Gate	Fan-out Gate
Dynamic Power Dissipation	Power Consumed by Internal Capacitance Transition Time Delay	Short-circuit Current

Figure 2.1: The Effect of Input Reordering in a 2-Input NAND Gate.

poured into the input B. The input capacitance of B, C_B, is calculated as

$$C_B = \frac{Q_B}{V_{DD}}. \tag{2.1}$$

Table 2.1 lists C_B under various conditions of other inputs and the initial voltage of internal nodes. The rightmost column(Ratio) indicates the ratio of the input capacitance under various conditions with respect to the value when both of inputs A and B are kept high. From Table 2.1, it can be observed that the signal value of the input C affects C_B strongly. In other words, C_B becomes small when the source of NB is floating from ground. Compared with the input C, the input A and the initial value of internal nodes have minor influence. Therefore the input capacitance is characterized under following two conditions; the condition that the source of the input transistor is connecting to ground, and the condition that the source is floating ground.

2.2.2 Reordered Gate

Internal capacitances in a reordered gate have an influence on the power dissipation, delay time, and transition time of the reordered gate. References [34, 35, 36, 37] discuss methods for power reduction by input reordering such that the number of charging and discharging the internal capacitances could be reduced. Let us take a 4-input NAND gate as an example to investigate how power dissipation and delay vary input by input. Table 2.2 lists the power dissipation(dissipated energy, rigorously), rise/fall delay times and transition times when the

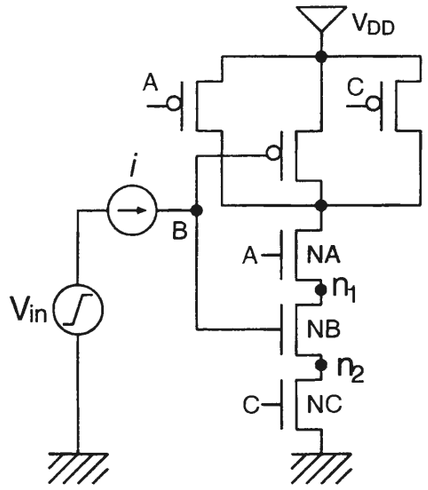


Figure 2.2: The Method of Measuring the Input Capacitance in a 3-Input NAND Gate.

output load capacitance is 60 fF and the transition time of the input signal is 0.4 ns. The gate is driven by input A or D, where input A is closest to the output and input D is closest to ground. The dissipated power(energy), rise delay time and rise transition time of input D are larger than those of input A by 79%, 69%, 78%, respectively.

Rise delay/transition times as well as fall delay/transition times show input pin dependencies. Even in transitions driven by a parallel-connected transistor(eg. output rise/fall for NAND/NOR gates), there exists the distinct input-pin dependency. This is because the amount of capacitances to be charged, which includes the internal capacitances between series-connected MOSFETs, depends on the location of the driving (input) pin. In Ref. [38], the input-pin dependency of the transitions driven by a parallel-connected transistor is neglected in its timing optimization process. The above example means that this simplification is not reasonable. The dependencies in both transitions make a delay optimization process not so straightforward, as described later.

Table 2.1: Input Capacitance of B under Various Conditions.

Input A	Input C	Node n_1	Node n_2	Input Capacitance(fF)	Ratio (%)
High	High	-	-	40	-
Low	High	High†	-	38	95
Low	High	Low	-	37	93
High	Low	-	High†	25	63
High	Low	-	Low	24	60
Low	Low	High†	High†	24	60
Low	Low	High†	Low	26	65
Low	Low	Low	Low	26	65

High : V_{DD} High† : $V_{DD} - V_{TH}$ Low : 0

Table 2.2: Typical Characteristics of a 4-Input NAND Gate.

	Pin A	Pin D	Pin D/Pin A
Power(pJ)	3.8	6.8	179%
Rise Delay(ns)	0.26	0.44	169%
Fall Delay(ns)	0.18	0.23	128%
Rise Transition Time(ns)	0.41	0.73	178%
Fall Transition Time(ns)	0.34	0.33	97%

2.2.3 Fan-out Gate

Input reordering of the reordered gate affects the power dissipation of a fan-out gate. This is because the reordering changes the transition time of the input signal of the fan-out gate, which leads to the change in the short-circuit current of the fan-out gate. If the transition time is short, the short-circuit power dissipation in the fan-out gate becomes small. This effect, however, is secondary compared to those of the fain-in gate and the reordered gate. Therefore a further discussion on fan-out gates is omitted.

2.3 Reordering Strategies

This section discusses the reordering strategies for each effect discussed in the previous section. The overall algorithm which combines the strategies for optimizing the total performance of the circuits will be shown in the next section.

2.3.1 Definitions

The primary input signal $x[n]$, a synchronized discrete-time logic signal, is defined as

$$x[n] = x(nT) = x(t)|_{t=nT}, \tag{2.2}$$

where n is an integer and T is the period of the system clock. The signal probability $P(x)$ and transition rate $R(x)$ are defined as follows.

$$P(x) = \lim_{k\to\infty} \frac{1}{k} \sum_{n=1}^{k} x[n]. \tag{2.3}$$

$$R(x) = \lim_{t\to\infty} \frac{n_x(t)}{t}, \tag{2.4}$$

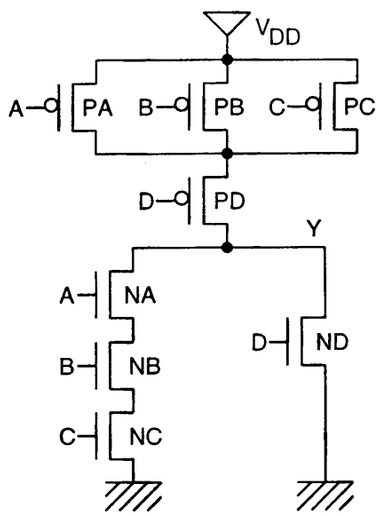


Figure 2.3: AOI31 Gate.

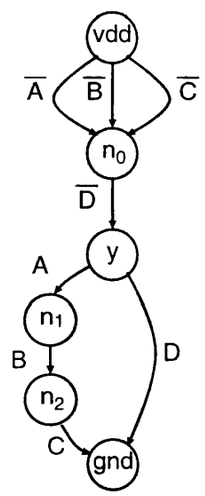


Figure 2.4: Graph of AOI31 Gate.

where $n_x(t)$ is the number of transitions of $x(t)$ between a time interval of length t, and $n_x(t)$ includes glitch transitions.

A static CMOS gate is represented as a directed acyclic graph (V, E)[36]. $V = \{n_0, \cdots, n_{p-1}, y, vdd, gnd\}$ is the set of nodes, where $(n_0, \cdots, n_{p-1})$ are the internal nodes of the gate, (y) is the output node and (vdd, gnd) are the power and ground nodes. E represents the $2q$ transistors (q of pMOS and q of nMOS) which connect the nodes in V. Each edge has a label representing the logical condition that the transistor corresponding to the edge is conductive. The graph of AOI31 gate(Fig. 2.3) is represented as Fig. 2.4. The boolean function H_{n_k} is defined such that it represents a logical sum of all possible paths from vdd to n_k, where each path is represented as a logical product of the label of the edges on the path. In the example of AOI31 gate, H_y is represented as $(\overline{A} + \overline{B} + \overline{C}) \cdot \overline{D}$. Similarly G_{n_k} is the boolean function that represents all possible paths from n_k to gnd. Boolean function $K_{n_k \to n_l}$ represents all possible paths from n_k to n_l.

2.3.2 Power Dissipation in Fan-in Gate

The strategy for reducing the power dissipated in fan-in gates is explained. To consider the effect that the input capacitance depends on other inputs, effective input capacitance is introduced as an integral average of the input capacitance. The input reordering changes the effective input capacitance. Therefore, if the input with high transition rate have smaller effective input capacitance, the power dissipation in the fan-in gate becomes smaller.

In Section 2.2.1, it is said that the input capacitance becomes small when the source of

the input n(p)-transistor is floating from ground(power supply), which is not accurate for a complex gate. The input capacitance A of AOI31 gate (Fig. 2.3) is examined as an example. Suppose the inputs (B, C, D) are (0, 1, 1). The source of transistor NA is not connected to ground through the transistors NB and NC. The input capacitance of A, however, does not look small. It is because the input transistor NA is connecting to ground through the transistor ND. Therefore in the case of nMOS, the input capacitance looks small when both the source and the drain are floating from ground. Similarly, in the case of pMOS, the input capacitance looks small when both the source and the drain are floating from power supply.

Now, the calculation of the effective input capacitance is explained. The boolean function GI_{X_i} is defined such that it represents the logical condition that the source of $\mathrm{NX_i}$ is connecting to ground, where $\mathrm{NX_i}$ is the n-transistor of input X_i.

$$GI_{X_i} = G_{n_k}, \tag{2.5}$$

where n_k is the node that corresponds to the source of $\mathrm{NX_i}$. Also the boolean function GI'_{X_i} is defined such that it represents the logical condition that the drain of $\mathrm{NX_i}$ is connecting to ground when $\mathrm{NX_i}$ is not conductive.

$$GI'_{X_i} = \sum_{n_j \in V_n} K_{n_j \to n_l} \cdot G_{n_j}|_{X_i=0}, \tag{2.6}$$

where n_l is the node that corresponds to the drain of $\mathrm{NX_i}$ and $\sum$ represents the boolean OR operation. V_n is a subset of V which consists of node (y) and all the nodes in the nMOS network. $(K_{n_j \to n_l} \cdot G_{n_j}|_{X_i=0})$ means the logical condition that node n_l is connected to ground via node n_j when $\mathrm{NX_i}$ is not conductive. Using Eqs. (2.5) and (2.6), the boolean function FG_{X_i}, which represents the condition that both the source and the drain of $\mathrm{NX_i}$ are floating from ground, is represented as follows.

$$FG_{X_i} = \overline{GI_{X_i}} \cdot \overline{GI'_{X_i}}. \tag{2.7}$$

Similarly, the boolean functions HI_{X_i} and HI'_{X_i} are defined.

$$HI_{X_i} = H_{n_m}, \tag{2.8}$$

where n_m is the node that corresponds to the source of $\mathrm{PX_i}$. $\mathrm{PX_i}$ is the p-transistor of input X_i.

$$HI'_{X_i} = \sum_{n_j \in V_p} K_{n_n \to n_j} \cdot H_{n_j}|_{X_i=1}, \tag{2.9}$$

where n_n is the node that corresponds to the drain of $\mathrm{PX_i}$. V_p is a subset of V which consists of node (y) and all the nodes in the pMOS network. The boolean function FH_{X_i}, which represents the condition that both the source and the drain of $\mathrm{PX_i}$ are floating from power supply, is represented as follows.

$$FH_{X_i} = \overline{HI_{X_i}} \cdot \overline{HI'_{X_i}}. \tag{2.10}$$

Using Eqs. (2.7) and (2.10), the effective input capacitance of X_i is represented as follows.

$$C_{X_i}^{eff} = C_{PX_i}P(\overline{FH_{X_i}}) + C_{PX_i}^{float}P(FH_{X_i}) + C_{NX_i}P(\overline{FG_{X_i}}) + C_{NX_i}^{float}P(FG_{X_i}), \quad (2.11)$$

where $C_{PX_i}^{float}(C_{NX_i}^{float})$ is the gate capacitance of $\mathrm{PX_i}(\mathrm{NX_i})$ when the source and the drain are floating from power supply(ground), and $C_{PX_i}(C_{NX_i})$ is the gate capacitance when the drain is connecting to power supply(ground).

In the case of input B of AOI31 gate(Fig. 2.4), GI_B, GI'_B, FG_B, HI_B, HI'_B and FH_B are represented as follows.

$$\begin{aligned} GI_B &= C, && (2.12)\\ GI'_B &= AD + 1\cdot 0 + 0\cdot C = AD, && (2.13)\\ FG_B &= \overline{C}\cdot\overline{AD}, && (2.14)\\ HI_B &= 1, && (2.15)\\ HI'_B &= \overline{D}\cdot(\overline{A}+\overline{C}) + (\overline{A}+\overline{C}) = \overline{A}+\overline{C}, && (2.16)\\ FH_B &= \overline{1}\cdot\overline{\overline{A}+\overline{C}} = 0. && (2.17) \end{aligned}$$

The effective input capacitance of B (C_B^{eff}) is represented as follows.

$$C_B^{eff} = C_{PB} + C_{NB}P(\overline{\overline{C}\cdot\overline{AD}}) + C_{NB}^{float}P(\overline{C}\cdot\overline{AD}). \quad (2.18)$$

From the above discussion, the inputs should be reordered so as to decrease PW_{input}(the sum of the power dissipation of charging the input capacitances).

$$PW_{input} = \frac{1}{2}V_{DD}^2\sum_{i=0}^{n-1}R(X_i)C_{X_i}^{eff}, \quad (2.19)$$

where n is the number of inputs of the reordered gate.

2.3.3 Power Dissipation in the Reordered Gate

CMOS complementary gates consist of series/parallel-connected MOSFETs. The internal capacitances between series-connected MOSFETs influence on power dissipation in the reordered gate. An effective estimation method of the number of transitions at each internal node is proposed[36]. This method is utilized for the power estimation of the reordered gate. Here the method is explained briefly according to Ref. [36].

The power consumption of node n_k produced by input X_i ($W_{n_k}|_{X_i}$) is represented as follows.

$$W_{n_k}|_{X_i} = \frac{1}{2}C_{n_k}V_{DD}(V_{DD}-V_{TH})\cdot R(n_k)|_{X_i}, \quad (2.20)$$

where C_{n_k} is the internal capacitance corresponding to node n_k. $R(n_k)|_{X_i}$ is the transition rate of the transitions caused by the input X_i at the node n_k. If there are no simultaneous transitions, $R(n_k)|_{X_i}$ is represented as follows.

Table 2.3: Delay Time of a 4-Input NAND Gate.

	Pin A	Pin D	Pin D/Pin A
Rise Delay(ns)	0.51	0.73	143%
Fall Delay(ns)	0.16	0.12	75%

$$R(n_k)|_{X_i} = R(X_i)\{P(\frac{\partial H_{n_k}}{\partial X_i})\overline{P(n_k)} + P(\frac{\partial G_{n_k}}{\partial X_i})P(n_k)\}. \quad (2.21)$$

The power dissipation of the reordered gate ($PW_{reordered}$) is represented as follows.

$$PW_{reordered} = \sum_{k=0}^{p-1}(\sum_{i=0}^{n-1}W_{n_k}|_{X_i}) + \frac{1}{2}C_{load}V_{DD}^2R(Y), \quad (2.22)$$

where C_{load} is the load capacitance and p is the number of internal nodes and n is the number of inputs of the reordered gate. Thus the inputs should be reordered so as to decrease $PW_{reordered}$.

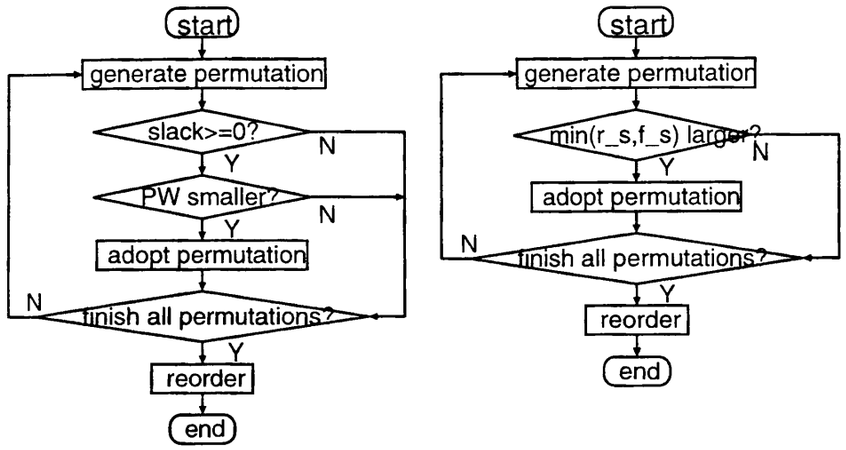
2.3.4 Delay

The delay of a gate differs not only input by input but also by the direction of output transition(rise/fall). Even in transitions driven by a parallel-connected transistor(eg. output rise/fall for NAND/NOR gates), there exists the input-pin dependency as seen in Table 2.2. Also the fall/rise delay of the pin with the smallest rise/fall delay is not necessarily the smallest. Table 2.3 shows the delay time of 4-input NAND gate when the output load capacitance is 60fF and the transition time of the input signal is 1.5ns. Table 2.3 is different from Table 2.2 in the condition of the input transition time. The rise delay of input A is smaller than the rise delay of D. However the fall delay of A is larger than the fall delay of D. Both rise and fall pin-to-pin delays for each input need to be considered instead of reducing them to a single pin-to-pin delay as is done in conventional timing optimization approaches [38, 39]. This implies that two delays(fall and rise delays) with each output should be associated.

In order to evaluate the contribution of each delay to the overall circuit delay, two slacks($rise\_slack, fall\_slack$) are calculated at each output and used as the measure of delay, where the slack is defined as the difference between the required arrival time and the latest arrival time[40]. For the delay optimization, the input order which makes $\min(rise\_slack, fall\_slack)$ the largest is chosen. This strategy is greedy to minimize the delay, and does not increase the delay of a critical path.

2.4 Optimization Algorithm

In the previous section, two strategies for power reduction and one strategy for delay reduction are shown. This section discusses an algorithm which combines the three strategies for the total performance optimization of the whole circuit.



(a) Power with delay constraint. (b) Delay.

Figure 2.5: Optimization Algorithm in Each Gate.

2.4.1 Optimization in Each Gate

This section shows an algorithm which optimizes a gate considering three strategies, that is to say, the strategy for the power reduction in the fan-in gates, the power reduction in the reordered gate, and the delay optimization for each gate.

First, the method to combine the power reduction strategy for the fan-in gates with that for the reordered gate is explained. Using two estimated power dissipations (PW_{input} and $PW_{reordered}$), it is considered that the input ordering which minimizes the total power $PW(= PW_{input} + PW_{reordered})$ is the best for low power. All the permutations are tried, and the one with the smallest PW is chosen. If the delay constraint is imposed on the power optimization, the slack is calculated for each permutation, and the ordering with the smallest PW and positive slack is selected. This flow is shown in Fig. 2.5(a).

In the case of delay optimization, $\min(rise\_slack, fall\_slack)$ is calculated for all the permutations, and the best order is selected. This flow is shown in Fig. 2.5(b).

2.4.2 Optimization of the Whole Circuit

For delay optimization, each gate is reordered with the strategy in Section 2.4.1, in a breadth-first search order starting from a gate with all the inputs driven by primary inputs. A reordering of a certain gate may change the slack of a gate not only in the fan-out direction but also in the fan-in direction. It is because that the required time of a gate in the fan-in direction changes and the rise slack and the fall slack of the gate change accordingly. So, the order which has been processed previously is not necessarily the best order. Even if all the gates in the circuit have been reordered once, there is a possibility that further delay reduction can be achieved. Therefore, the delay optimization requires iterative optimization. The delay

optimization loop finishes when the delay of critical path can not be decreased. In the case of power optimization, the algorithm in Section 2.4.1 is applied to each gate once, assuming input reordering does not change the transition rate. In the case of delay and power optimization, delay optimization is executed first for minimizing the critical path delay. After that, power optimization is processed under the delay constraint as shown in Fig. 2.5(a).

2.5 Experimental Results

In this section, the results of performance optimization by input reordering is shown. All of experiments in this section are achieved with the condition below. Process parameters for a commercial 0.7μm process is used. The power dissipation is evaluated by an event-driven transistor-level power simulator with the option that enables to consider the dependence of input capacitance[41]. Input patterns are randomly generated with a signal probability of 0.5 and with a transition density of 0.5, where transition density represents the average number of transitions per cycle[42]. The number of applied patterns is 100, which is the adequate number for the power estimation at circuit level [43]. The circuits are operated synchronously. The cycle time of input patterns is 20ns, which is the sufficient time for all benchmark circuits to finish the behavior. The transition rate R at each gate is computed by logic simulation, and the signal probability P is calculated using SBDD(shared binary decision diagram) [1]. The circuits used for the experiments are taken from ISCAS85 and LGSynth93 benchmark sets(See Table 2.4). The circuits are synthesized and mapped by a commercial logic synthesis tool. The target library includes basic gates and complex gates and selectors. These gates are the standard cells generated by P2Lib[44].

Table 2.4 lists the result of power optimization without delay optimization. The columns under "Initial" show the power dissipation(delay) of the initial circuits. The circuits are optimized by input reordering with the following three strategies.

A: The strategy which considers the dissipated power in the fan-in gates and the reordered gate (proposed).

B: The strategy which considers the dissipated power only in the reordered gate (equivalent to Ref. [36]).

C: The strategy which maximizes the power dissipation using the proposed method.

The columns of "A" and "B" under "Reduction" represent the percentage of the power reduction ($= \frac{Initial-A(or\,B)}{Initial} \times 100(\%)$). The column "Diff." explains the percentage of the difference between the largest and the smallest power dissipations ($= \frac{C-A}{A} \times 100(\%)$). The column "CPU Time" lists a CPU time for reordering on a Sun Ultra 2. It does not include the time to calculate transition rate by logic simulation.

From Table 2.4, it can be seen that power dissipation of all circuits is reduced by the proposed method. The "Diff." column indicates that there is a possibility of reducing power

[1] BDD Manipulator ver 6.03 : Copyright 1992 Kyoto University (by Shin-ichi MINATO).

dissipation by 22.5% maximum. The proposed method (Column "A") reduces power dissipation by 5.9 % on average and by 12.9 % maximum, whereas a conventional method, which considers the dissipated power only in the internal capacitances of the reordered gate, reduces power dissipation by 3.6 % on average and by 10.4 % maximum. The power optimization without delay optimization does not affect delay so much.

In Table 2.5, the result of power optimization with delay optimization is shown. The proposed method reduces delay by 6.7 % and power dissipation by 5.3 % on average.

2.6 Conclusion

This chapter discusses an improved method for power optimization of CMOS gates by input reordering. The dependence of input capacitance on the signal values of other inputs, as well as the possibility of charging/discharging internal capacitances, is utilized for the power reduction. The effect of the method is demonstrated experimentally using 30 benchmark circuits in a 0.7 μm CMOS technology. The average reduction of power dissipation is 5.9 %. By input reordering there is a possibility that power dissipation is reduced by 22.5% maximum. In the case of delay and power optimization, the proposed method improves delay by 6.7 % and power dissipation by 5.3% on average. Although the amount of improvement in power and delay is not drastic, input reordering can provide a steady improvement with almost zero penalty.

Table 2.4: Power Optimization without Delay Optimization.

Circuit	Power Dissipation				Delay		CPU	NO.
	Initial	Reduction(%)		Diff.	Initial	Reduc-	Time	of
	(mW)	A†	B‡	(%)	(ns)	tion(%)	(s)	Gate
sao2	2.69	3.5	0.4	13.0	4.25	3.2	0.7	100
my_adder	4.60	6.3	6.2	5.8	11.8	0.1	42.1	112
c432	5.57	12.9	9.9	19.4	10.2	4.0	6.2	112
apex7	4.79	5.6	3.8	9.9	3.90	3.3	0.7	135
clip	5.75	3.6	2.4	8.4	4.45	0.2	0.7	154
term1	5.94	4.5	1.6	7.2	3.76	-2.0	0.7	171
example2	4.46	5.2	1.6	22.3	4.02	3.5	1.0	172
c499	3.92	1.9	0.3	10.9	4.63	0.7	55.7	176
alu2	8.87	7.4	6.5	9.1	10.5	-0.6	1.3	197
x4	7.30	9.3	10.3	22.5	3.77	-0.7	2.9	201
dule2	4.00	4.4	0.7	17.1	5.08	3.9	0.8	210
c1908	8.48	10.2	8.0	20.9	10.1	3.3	503.2	249
i9	17.8	6.1	6.8	7.1	4.19	1.5	0.5	306
i7	15.9	5.3	5.4	7.8	3.61	-1.1	0.5	314
c1355	12.5	7.4	4.7	10.9	8.48	3.6	160.2	326
e64	4.84	5.1	-1.1	12.5	4.42	-13.7	0.8	327
table5	5.38	7.7	0.4	19.5	6.22	3.3	1.2	383
apex6	15.0	5.7	6.5	10.9	4.44	3.9	7.1	391
dalu	15.0	7.8	6.0	15.0	7.58	-3.6	6.6	407
x3	14.5	2.5	1.3	5.3	3.26	-4.1	0.7	447
table3	5.71	7.3	0.0	17.3	6.22	1.0	1.1	454
frg2	16.0	4.4	2.5	8.8	5.74	-0.4	2.3	476
i8	25.8	7.8	7.4	7.2	7.71	1.7	5.0	535
c3540	29.1	3.4	2.4	7.0	12.0	0.8	9.5	585
apex3	10.8	7.2	2.8	15.3	6.68	1.0	1.3	734
ex5p	14.2	5.6	5.6	10.4	7.12	5.7	1.1	935
alu4	29.0	3.5	0.6	10.7	6.91	2.4	5.8	937
apex2	25.6	1.7	-0.7	13.6	8.09	-2.5	16.8	1253
seq	27.7	5.1	1.5	13.3	8.30	5.1	6.1	1370
des	73.9	7.5	4.5	13.0	7.64	3.4	29.8	1718
Average	-	5.9	3.6	12.4	-	0.9	-	-

A† : Proposed Method B‡ : Conventional Method

Table 2.5: Delay and Power Optimization.

Circuit	Delay Reduction(%)	Power Reduction(%)	Time (s)
sao2	11.2	2.4	0.9
my_adder	0.1	6.3	26.0
c432	9.9	7.0	20.4
apex7	9.9	3.9	0.9
clip	5.7	3.1	1.1
term1	9.6	2.4	0.9
example2	7.9	5.2	1.2
c499	7.6	0.7	30.7
alu2	4.4	8.1	1.5
x4	4.4	5.2	2.9
dule2	11.9	3.4	1.2
c1908	8.1	10.0	495.3
i9	4.4	6.4	0.9
i7	4.3	4.8	1.1
c1355	9.1	6.5	115.1
e64	2.9	4.9	1.5
table5	8.3	7.3	1.9
apex6	7.0	5.4	10.5
dalu	7.1	8.2	10.0
x3	7.3	2.9	1.5
table3	4.9	7.1	2.2
frg2	4.6	3.7	2.6
i8	1.9	9.3	7.6
c3540	6.4	2.6	21.2
apex3	4.5	8.1	3.1
ex5p	8.2	6.2	5.2
alu4	7.6	3.1	9.3
apex2	7.8	2.1	21.7
seq	6.3	4.4	10.6
des	8.4	7.5	40.9
Average	6.7	5.3	-

Chapter 3

Gate Sizing for Glitch Power Reduction

This chapter discusses a method for power optimization that considers glitch reduction by gate sizing based on the statistical estimation of glitch transitions. The proposed method reduces not only the amount of capacitive and short-circuit power consumption but also the power dissipated by glitches. The effectiveness of the proposed method is verified experimentally using 10 benchmark circuits with a 0.6 μm standard cell library. The proposed method reduces power dissipation from the minimum-area circuits further by 10.4% on average and 16.2% maximum. It is also verified that the proposed method is effective under manufacturing variation.

3.1 Introduction

In the various stages of VLSI design, many techniques for power reduction have been proposed, such as supply-voltage scaling[29, 30], technology mapping for low power[31], input reordering[45, 37], gate sizing[4, 46, 47, 48], and so on. This paper focuses on gate sizing which is an effective method not only for delay optimization[1] but also for power optimization. The circuit under optimization is a CMOS combinational circuit designed in a synchronous design style.

The dynamic power dissipation, which is the dominant source of power dissipation, is directly related to the number of signal transitions in a circuit. A signal transition can be classified into two categories; a functional transition and a glitch. It is well known that glitches occupy a considerable amount of the signal transitions in a circuit. Reference[49] indicates that the glitch power dissipation accounts for 20% to 70%, and Ref.[43] tells 7% to 43%. Also glitches are extremely sensitive to delay characteristics[50]. Therefore glitch reduction by optimizing delay characteristics is a reasonable approach for power reduction.

This chapter proposes a gate sizing method considering glitch reduction for low power design. Conventional approaches for power reduction optimize the amount of capacitive load[4, 48] or the amount of capacitive load and short-circuit current[47, 51] based on the transition activity information obtained beforehand. Recently, some glitch power reduction methods are proposed[52, 53]. In order to eliminate glitches completely, the authors[52]

adjust the gate delay time and insert buffers such that the time difference between the latest arrival time and the earliest arrival time at each gate becomes smaller than its gate delay time. In practical circuits, the time difference between the latest and earliest arrival time is much larger than the gate delay time at most of gates. Also the gate delay is not allowed to be excessively long, because the transition time constraints are usually given for maintaining the accuracy in timing analysis and the hot-carrier reliability. The cost, i.e. the number of inserted buffers, required to remove glitches entirely is not small, and hence the power dissipation of the circuit optimized to eliminate glitches completely is not minimum. Therefore Ref. [52] can not minimize the power dissipation. Reference [53] proposes a glitch power reduction method by gate freezing. Gate freezing replaces some existing gates with "F-Gates" that do not propagates glitches according to the given control signal. In this method, the timing of the control signal is critical and essential not only to reduce glitches but also to ensure the correct behavior of a circuit. If the timing of the control signal is varied by manufacturing variability or the timing calculation error, the functional signals may not propagate through the circuit. Therefore gate freezing[53] requires extensive verification to avoid functional failures caused by delay fluctuation. The proposed method reduces not only the amount of capacitive and short-circuit power consumption but also the power dissipated by glitches explicitly with an improved glitch estimation technique. The proposed method reduces glitch power dissipation by gate sizing, and hence the correct functional behavior is guaranteed against delay fluctuation caused by manufacturing variability and delay calculation error.

The proposed optimization method consists of two techniques; a statistical estimation method of glitch activities and an optimization algorithm for gate resizing. For the estimation of glitch activities, glitches are classified into two classes; generated glitches and propagating glitches. As for the generated glitches, a statistical estimation method proposed by Lim and Soma[54] is adopted. The propagating glitches, however, are not considered in their method, and therefore a statistical estimation method is developed. The optimization algorithm has been designed to have the ability of escaping from a bad local solution while keeping small computational costs.

In real circuits, there exist statistical perturbations of circuit parameters such as skew fluctuations and variabilities in gate delay, which may affect glitch activities and thereby cannot be neglected. Also, not all glitches have full-swing transitions. Treating all glitches as full-swing transitions may cause an excessive overestimation of glitch power dissipation. This chapter proposes a practical power optimization method considering actual phenomena, such as skew fluctuations and partial-swing transitions.

This chapter is organized as follows. Section 3.2 discusses the statistical glitch estimation method considering propagating glitches, skew fluctuations and partial-swing transitions. Section 3.3 explains the optimization algorithm of gate resizing. Section 3.4 shows some experimental results of the proposed method. Finally Section 3.5 concludes the discussion.

3.2 Statistical Glitch Estimation

This section explains an estimation method for glitch activities based on a statistical approach. Glitches can be separated into the following two components.

generated glitches: the glitches that are generated by functional (non-glitch) transitions.

propagating glitches: the glitches that are generated previously at a gate in the fan-in direction and propagate through the gate.

As for the generated glitches, a statistical estimation method is proposed by Lim and Soma[54]. However, the effect of propagating glitches is not taken into account. Some part of the generated glitches may be immediately blocked by the fan-out gates. Other part, however, will propagate through the circuit until they are suppressed or reach to primary outputs. Therefore the effect of the propagating glitches cannot be neglected.

The voltage swing of glitches is not always V_{DD}. The energy dissipated by charging and discharging the load capacitance is proportional to the voltage swing. Treating all glitches as full-swing transitions cause an overestimation of the power dissipated by glitches. Therefore the estimation method of the generated glitches[54] is improved such that the power dissipated by partial-swing transitions can be considered.

In real circuits, there exist uncertainties in delay characteristics, which may spoil the effect of power optimization. For example, after a clock distribution tree is designed, the skew time at each flip-flop(latch) can be estimated. However, the estimated skew time has some errors. Also, the skew time fluctuates owing to the statistical variation of the transistor characteristics and the wire capacitance. The skew fluctuation affects the transition timing at the primary inputs in combinational circuits, and consequently influences the glitch generation. Therefore the estimation method that can consider skew fluctuations is contrived. This consideration increases the tolerance of glitch reduction to actual phenomena in real circuits.

3.2.1 Preparations

The primary input signal $x[n]$, a synchronized discrete-time logic signal is defined as

$$x[n] = x(nT) = x(t)|_{t=nT}, \tag{3.1}$$

where n is an integer and T is the period of the system clock. The signal probability $P(x)$ and the transition density $D(x)$ are defined as follows[42].

$$P(x) = \lim_{k\to\infty} \frac{1}{k} \sum_{n=1}^{k} x[n], \tag{3.2}$$

$$D(x) = \lim_{k\to\infty} \frac{1}{k} \sum_{n=1}^{k} |x[n] - x[n-1]|_{x[0]=x_0}, \tag{3.3}$$

where x_0 is an initial logic value. The switching probabilities $P^{00}(x)$, $P^{01}(x)$, $P^{10}(x)$ and $P^{11}(x)$ are the probabilities that the signal of gate x changes as $0 \rightarrow 0$, $0 \rightarrow 1$, $1 \rightarrow 0$, $1 \rightarrow 1$, respectively. These probabilities have the following relations.

$$P^{00}(x) + P^{01}(x) + P^{10}(x) + P^{11}(x) = 1, \tag{3.4}$$

$$P^{01}(x) = P^{10}(x) = \frac{D(x)}{2}, \tag{3.5}$$

$$P^{11}(x) + P^{10}(x) = P(x). \tag{3.6}$$

Transition rate $R(x)$ is defined as

$$R(x) = \lim_{t \rightarrow \infty} \frac{n_x(t)}{t}, \tag{3.7}$$

where $n_x(t)$ is the number of transitions of $x(t)$ between a time interval of length t.

In order to consider short-circuit power dissipation, a power estimation method based on look-up tables is utilized. In this method, the total power dissipation PW, including short-circuit power dissipation, is represented as follows.

$$PW = \frac{1}{2} \sum_{i}^{n} PW_{table}(i) R(i), \tag{3.8}$$

where n is the number of gates and $PW_{table}(i)$ is the energy that is consumed at the gate i when the output changes. The values of $PW_{table}(i)$ are given by look-up tables which includes the power dissipated by the short-circuit current. The look-up tables are two-dimension tables with load capacitance and input transition time as variables and they are characterized beforehand by circuit simulation. Equation (3.8) is used as the objective function of power optimization.

Path delays are derived using a static timing calculation method. As for gate delay calculation at each gate, two dimensional look-up tables with capacitive load and input transition time as parameters is used. The look-up tables of the gate delay and the transition time of the output signal are characterized by circuit simulation.

3.2.2 Previous Work on Generated Glitch

First, the estimation method for generated glitches[54] is explained. The condition for glitch generation is to hold the following two conditions simultaneously(Fig. 3.1).

Condition 1: The input pattern ω_k is the pattern that can cause glitches.

Condition 2: The interval time ζ between successive transitions at different inputs is larger than the gate delay time τ.

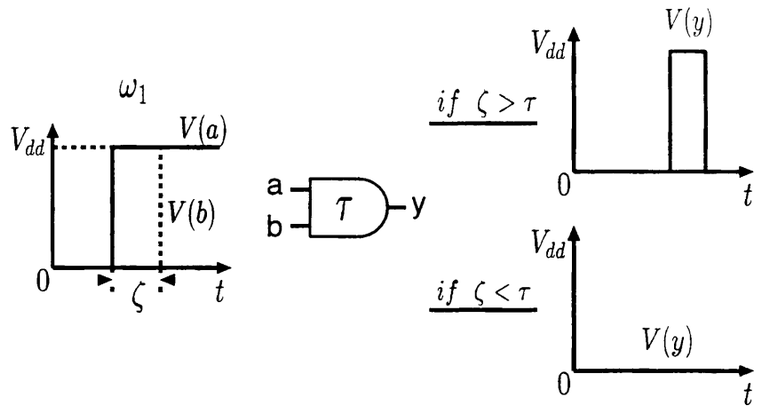


Figure 3.1: An Input Pattern and Condition for Glitch Generation in a 2-Input AND Gate.

The probability satisfying Condition 1 and the probability satisfying Condition 2 are calculated separately. The pattern probability $P_{patt}(\omega_k)$ is the probability that the input pattern ω_k occurs. The generation probability $P_{gen}(\omega_k)$ is the probability that the input pattern ω_k satisfies Condition 2, and is represented as follows:

$$P_{gen}(\omega_k) = \int \int_{A_k} f(\alpha) f(\beta) d\alpha d\beta, \tag{3.9}$$

where α and β are the arrival times of the respective signals in ω_k, f is the distribution function that represents the number of transitions as a function of arrival time. A_k is the area that satisfies Condition 2 in the $\alpha - \beta$ space(Example, Fig. 3.2). In Fig. 3.2, parameters $\alpha_{min}(\beta_{min})$ and $\alpha_{max}(\beta_{max})$ represent the earliest and the latest arrival times respectively. Parameter $\tau_\alpha(\tau_\beta)$ represents the gate delay time of signal $\alpha(\beta)$. Using P_{patt} and P_{gen}, generated glitch rate $R_{gen}(i)$ is represented as follows.

$$R_{gen}(i) = f_{clk} \cdot \sum_{k} \{P_{gen}(\omega_k) \cdot P_{patt}(\omega_k)\}, \tag{3.10}$$

where f_{clk} is the clock frequency.

3.2.3 Propagating Glitch

The propagating glitch rate $R_{prop}(x)$ is defined as follows:

$$R_{prop}(x) = \lim_{t \rightarrow \infty} \frac{n_{prop\_x}(t)}{t}, \tag{3.11}$$

where $n_{prop\_x}(t)$ is the number of propagating glitches at the gate x between a time interval of length t. From the definitions, the total transition rate R can be represented using D, R_{gen},

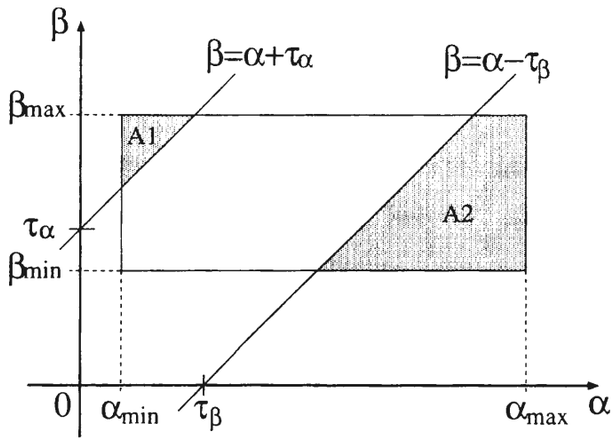


Figure 3.2: Surface Integral Area of the Distribution Function f.

R_{prop}, f_{clk} as follows:

$$R(x) = f_{clk} \cdot D(x) + 2 \cdot \{R_{gen}(x) + R_{prop}(x)\}. \tag{3.12}$$

The multiplication factor of two in the second term comes from that a single glitch causes two transitions.

Now, an estimation method of the propagating glitch rate R_{prop} is explained. Here, the disappearance of the glitches whose time widths are shorter than the delay of the propagating gate is ignored. If the inputs of a gate have no correlation with each other and there is a sufficient time interval between the input transitions, the following equation holds at any gates[42].

$$R(y) = \sum_{i=1}^{n} P(\frac{\partial y}{\partial x_i}) R(x_i), \tag{3.13}$$

where x_i is the i-th input of the gate, y is the output and n is the total number of inputs. From the definition of R_{prop}, if the glitches at the inputs have no correlation and have sufficient time interval between the transitions, R_{prop} can be represented as follows.

$$R_{prop}(y) = \sum_{i=1}^{n} P(\frac{\partial y}{\partial x_i}) \cdot \{R_{gen}(x_i) + R_{prop}(x_i)\}. \tag{3.14}$$

In the case of 2-input AND gate, Eq. (3.14) is represented as follows.

$$\begin{aligned} R_{prop}(y) \;=\; & P(b) \cdot \{R_{gen}(a) + R_{prop}(a)\} \\ & + P(a) \cdot \{R_{gen}(b) + R_{prop}(b)\}. \end{aligned} \tag{3.15}$$

Using Eq. (3.6), Eq. (3.15) is transformed to:

$$\begin{aligned} R_{prop}(y) = & \{P^{11}(b) + P^{10}(b)\} \cdot \{R_{gen}(a) + R_{prop}(a)\} \\ & + \{P^{11}(a) + P^{10}(a)\} \cdot \{R_{gen}(b) + R_{prop}(b)\}. \end{aligned} \tag{3.16}$$

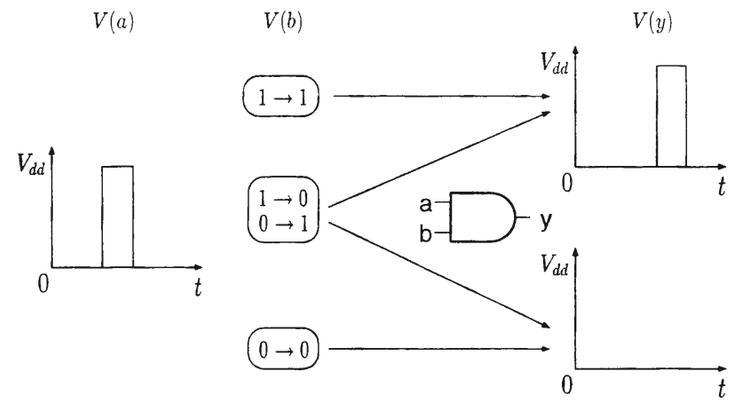


Figure 3.3: The Condition that Allows a Glitch Propagating through a 2-input AND Gate in the Case that the Gate Delay is Smaller than the Glitch Width.

Equation (3.14) assumes that there is a sufficient time interval between the transitions, so this equation may overestimate propagating glitches. There is a possibility that the overestimation of propagating glitches at each gate causes an excessive overestimation along with the signal propagation. Therefore the lower bound of propagating glitches should be estimated. Please consider the situation that a glitch comes from the input A in a 2-input AND gate (Fig.3.3). If the input B retains high, the glitch propagates through the gate. If the input B keeps low, the glitch never propagates through the gate. But if there is a transition at the input B, glitch propagation through the gate depends on the timing of the transition. In order to take the lower bound of the estimation, the timing-dependent glitch propagation is neglected. Therefore the estimation of the propagating glitch rate becomes:

$$\begin{aligned} \min\{R_{prop}(y)\} = & P^{11}(b) \cdot \{R_{gen}(a) + R_{prop}(a)\} \\ & + P^{11}(a) \cdot \{R_{gen}(b) + R_{prop}(b)\}. \end{aligned} \tag{3.17}$$

The above equation is obtained by setting P^{10} in Eq. (3.16) to be zero. Similar discussion can be made for other kinds of gates. Therefore the lower bound of the propagating glitch rate R_{prop} is calculated from Eq. (3.14) as:

$$R_{prop}(y) = \sum_{i=1}^{n} \{R_{gen}(x_i) + R_{prop}(x_i)\} \cdot P(\frac{\partial y}{\partial x_i})\Big|_{P^{10}=P^{01}=0}. \tag{3.18}$$

3.2.4 Partial-Swing Transitions

The energy dissipated by charging and discharging the load capacitance C is proportional to the voltage swing. When the voltage swing is $V_{DD}/2$, the dissipated energy which is represented as $C \cdot \frac{V_{DD}}{2} \cdot V_{DD}$ is the half of the energy of a full-swing transition. Treating

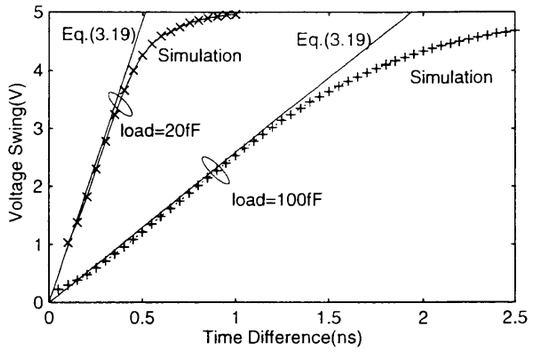


Figure 3.4: Relationship between the Swing Voltage and the Difference of the Arrival Time in 2-input NAND Gate.

a partial-swing transition as a full-swing transition causes an overestimation of the energy dissipated by glitches. Therefore an approach is devised such that a partial-swing transition is converted into an equivalent fraction of a full-swing transition based on the dissipated energy. For example, a transition that the voltage swing is $V_{DD}/2$ is regarded as 0.5 transition.

Fig. 3.4 shows the relationship between the voltage swing V_{SW} and the difference of the arrival time$\gamma(=\alpha-\beta$ or $\beta-\alpha)$ in a 2-input NAND gate. The relationship under two output load conditions is examined by circuit simulation, and it is approximated as a linear function.

$$V_{SW} = \begin{cases} \frac{V_{DD}}{2\tau}\gamma & 0 \le \gamma \le 2\tau \\ V_{DD} & \gamma > 2\tau \end{cases} . \tag{3.19}$$

Similarly, in the other gates, such as multi-stage gates, the relationship between V_{SW} and τ is examined, and it is approximate as a linear function.

Using this conversion, Eq. (3.9) can be improved as follows.

$$P_{gen}(\omega_k) = \int\int f(\alpha)f(\beta)h(\alpha,\beta)d\alpha d\beta, \tag{3.20}$$

$$h(\alpha,\beta) = \frac{V_{SW}(\alpha,\beta)}{V_{DD}}. \tag{3.21}$$

3.2.5 Distribution Function

The rigorous derivation of the distribution function f requires two processes. The first process is to search all paths and calculate the delay of each path. The complexity of this process is $O(n^d)$ where n is the average fan-in and d is the maximum circuit depth. The second process is to evaluate the activating probability of each path. This process requires the derivation of the sensitization conditions for all the paths, and hence overall complexity is practically infeasible. Therefore a simple and reasonable shape should be assumed for the distribution function f.

A possible shape might be a normal distribution. However, the estimation of the mean and the deviation of the normal distribution is not simple. Also, as will be shown in Section 3.4.1, the assumption of the normal distribution is not always reasonable in a real circuit. Here using an uniform distribution is proposed. The validity of this assumption will be examined experimentally in Section 3.4.1. The uniform distribution function f is represented as follows:

$$f(t) = \frac{1}{\alpha_{max}-\alpha_{min}} \cdot \{U(t-\alpha_{min}) - U(t-\alpha_{max})\}, \tag{3.22}$$

where α_{max} is the latest arrival time and α_{min} is the fastest arrival time.

When the distribution function f is uniform, $h(\alpha,\beta)$ of Eq. (3.21) can be transformed as follows.

$$h(\alpha,\beta) = \begin{cases} U(\alpha-\beta-\tau'_\beta) & 0 \le \alpha-\beta \\ U(\beta-\alpha-\tau'_\alpha) & 0 \le \beta-\alpha \end{cases}, \tag{3.23}$$

where τ' is derived from the following equation.

$$\int U(\gamma-\tau')d\gamma = \int h(\gamma)d\gamma. \tag{3.24}$$

In the case of Eq.(3.19), $\tau'_\alpha(\tau'_\beta)$ is represented as $\tau_\alpha(\tau_\beta)$.

Using Eqs.(3.22) and (3.23), Eq.(3.20) can be transformed as follows(Fig.3.5).

$$P_{gen}(\omega_k) = \int\int_{A'_k} f(\alpha)f(\beta)d\alpha d\beta \tag{3.25}$$

$$= \frac{\text{area}(A'_k)}{(\alpha_{max}-\alpha_{min})(\beta_{max}-\beta_{min})}, \tag{3.26}$$

where area(A'_k) represents the shaded area in Fig. 3.5.

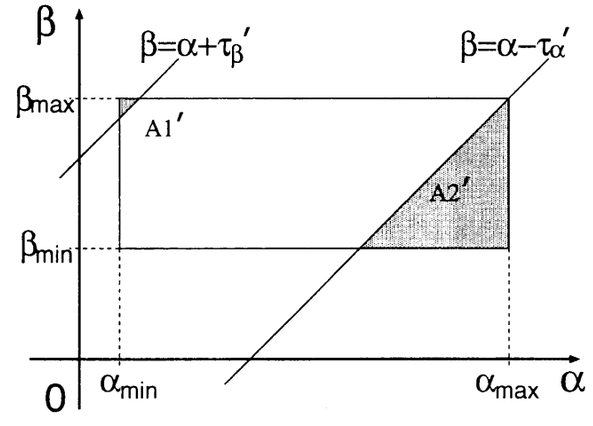


Figure 3.5: Surface Integral Area of the Distribution Function f Considering Partial-Swing Transitions.

3.2.6 Skew Fluctuation

After a clock distribution tree is designed, the skew time at each flip-flop(latch) can be estimated. However, the estimated skew time has certain amount of estimation errors. Also, the skew time varies due to manufacturing variability. Therefore skew fluctuation should be considered in glitch estimation. It is assumed that the distribution of the skew time is normal(μ, σ) and μ is the estimated skew time. Normally distributed skew at each primary input appears as the skew in the arrival time at the input of each gate. The distribution of the skew is well approximated by normal[55]. Hence $P_{gen}(\omega_k)$ under skew fluctuation is approximated as the weighted average over five sampling points.

$$\begin{aligned} P_{gen}(\omega_k) &= 0.404 \int\int_{A'_k} f(\alpha)f(\beta)d\alpha d\beta \\ &+ 0.149 \int\int_{A'_k} f(\alpha-\sigma)f(\beta-\sigma)d\alpha d\beta \\ &+ 0.149 \int\int_{A'_k} f(\alpha-\sigma)f(\beta+\sigma)d\alpha d\beta \\ &+ 0.149 \int\int_{A'_k} f(\alpha+\sigma)f(\beta-\sigma)d\alpha d\beta \\ &+ 0.149 \int\int_{A'_k} f(\alpha+\sigma)f(\beta+\sigma)d\alpha d\beta. \end{aligned} \tag{3.27}$$

3.3 Optimization Algorithm for Power Reduction

Given the estimation of glitch transitions, a good measure of overall power dissipation is obtained. Discrete (cell-based) gate sizing is executed for power optimization of a CMOS combinational circuit using the estimation method. This section explains the optimization algorithm for power reduction.

A heuristic algorithm that has both the merit of rapid convergence and the ability to get out of a bad local solution is developed. Here, the algorithm under delay and transition time constraints is explained. A flow-chart of the algorithm is shown in Fig. 3.6.

Optimize delay: The circuit is optimized by a similar algorithm to this power optimization until the delay constraints are satisfied. The detail is explained later.

Calculate sensitivity: At each gate, the sensitivity of the objective function Eq.(8) is evaluated both for sizing-up and sizing-down operations. If a sizing operation violates delay constraints or transition time constraints, the sensitivity is not calculated and the operation is eliminated from sizing candidates.

Resize: Gates are selected according to the sensitivity and they are resized. The number of the gates resized simultaneously is at most *Max_Change*.

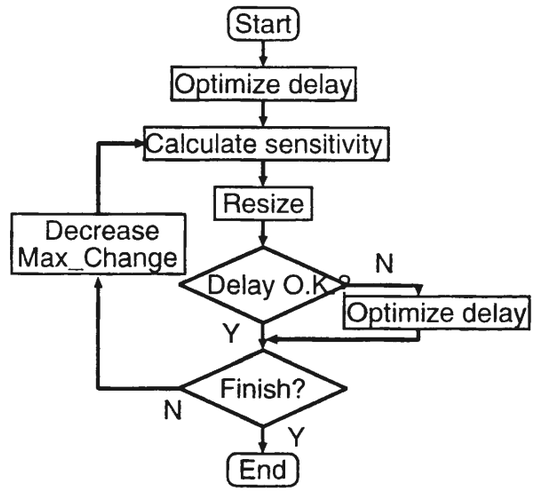


Figure 3.6: The Power Optimization Algorithm under Delay Constraints.

Delay O.K.?: There is a possibility that timing violation occurs because at most *Max_Change* gates are resized at once. It is judged whether the delay constraints are satisfied or not.

Finish?: If the iteration count goes over a pre-defined value *Max_Iteration*, or if no gates are resized, the optimization procedure finishes.

Decrease *Mac_Change*: *Max_Change* is reduced by a factor of *Reduce_Rate*.

The uncertainty of gate delay is aggravated by a signal that has an excessive transition time, i.e. the calculation error of gate delay increases and gate delay becomes sensitive to manufacturing variability. Also the long transition time deteriorates hot-carrier reliability. Therefore the transition time of the signals should be restricted. The sensitivity is calculated when the sizing does not violate the constraints of transition time. This restriction of transition time helps to maintain the accuracy of timing analysis and the reliability.

In the case of power optimization, the objective function is Eq. (3.8). As Eq. (3.8) includes short-circuit power dissipation, the power optimization considering overall power dissipation can be executed. Since at most *Max_Change* gates are resized at a time, there is no guarantee that the overall resizing results in the improvement of the objective function. The evaluated sensitivity for each gate is only valid for single resizing of the corresponding gate. This simultaneous resizing is regarded as a perturbation to the circuit. The amount of perturbation is reduced as the number of *Max_Change* is decreased through the iteration.

In the beginning of the optimization, i.e., when *Max_Change* is large, many gates are resized simultaneously. In this case, the amount of perturbation is large, and solution space is expected to be explored globally. Parameter *Max_Change* is gradually reduced at the

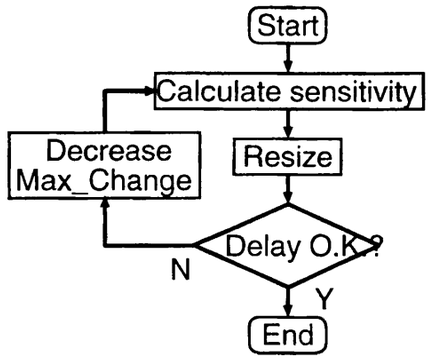


Figure 3.7: The Delay Optimization Algorithm Used in Power Optimization.

rate of *Reduce_Rate*, and the amount of perturbation decreases. The gradual reduction of *Max_Change* has a similar role to the temperature reduction in simulated annealing. The ratio of reduction can control the speed of convergence and the search area of solutions. At the final stage, *Max_Change* becomes small and this algorithm behaves like a greedy algorithm. A greedy algorithm is suitable for finding a local optimal solution, which merit is exploited in the proposed algorithm at the final stage. With the help of the perturbation and the greediness, it can be expected to reach to a good solution quickly. Tuning the parameters *Max_Iteration*, *Max_Change*, and *Reduce_Rate*, the amount of perturbations and convergence speed can be adjusted. Consequently the computation time and quality of the solution can be controlled.

The delay optimization executed in the power optimization is similar with the power optimization algorithm. Fig. 3.7 shows the flow of the delay optimization. First, the sensitivity of the circuit delay is evaluated for both size-up and size-down operations. In the sensitivity calculation, the timing information is updated at the gates in the downstream cone from the gates that drive the resized gate. Then the gates to be resized are chosen based on the sensitivity, and they are resized. The number of the gates resized simultaneously is at most *Max_Change*. If the delay constraint is satisfied, or if no gates are resized, the optimization finishes. Otherwise, the value of *Max_Change* is reduced and go back to the sensitivity calculation. The parameters *Max_Iteration*, *Max_Change*, and *Reduce_Rate* are assigned separately for delay and power optimization.

3.4 Experimental Results

This section shows some experimental results. First, the accuracy of the proposed glitch estimation method is verified experimentally. Next, power optimization results are demonstrated and the effectiveness of the proposed method is verified. Finally, it is shown that the proposed method can reduce glitches under the fluctuation of skew times and wire capacitances.

The circuits used for the experiments are an ALU in a DSP for mobile phone[67]

(dsp_alu) and the circuits included in ISCAS85 and LGSynth93 benchmark sets(C3540, ex5p, misex3, alu4, C5315, i10, seq, C7552, des). These circuits are synthesized and mapped by a commercial logic synthesis tool[56] such that the area is minimized under the transition time constraint of 1.5ns. The target library is a standard cell library used for actual fabrication in a 0.6 μm process with three metal layers. The library includes basic and complex gates. Buffer and inverter have six varieties in the driving strength and other gates have three varieties. The transition density D and signal probability P at each gate are calculated by logic simulation. The power dissipation is evaluated by a commercial transistor-level power simulator[41]. Input patterns are randomly generated with a signal probability of 0.5. The number of applied patterns is 1000, which is the adequate number for the power estimation at circuit level[43]. The cycle time of the input patterns is 100ns, which is a sufficient time for all benchmark circuits to finish the behavior. The constants for power optimization *Max_Iteration*, *Reduce_Rate* and initial *Max_Change* are set to 50, 0.90, 0.4$\times$(number of gates), respectively. The objective function is Eq. (3.8) which represents dynamic power dissipation including short-circuit power dissipation. The proposed method can therefore optimize circuits considering overall power dissipation.

3.4.1 Distribution Function

The validity of the uniform distribution function f , which is used for generated glitch estimation, is examined. The uniform distribution and the normal distribution are compared with the distribution that is extracted from the logic simulation.

The distribution function $f_{simulated}(t)$ is constructed from the logic simulation results. The number of the applied input pattern is 10000, and C3540 and des circuits are used for the experiment. The mean and the deviation are extracted from $f_{simulated}(t)$ and the normal distribution function $f_{normal}(t)$ is built. The uniform distribution function (Eq. 3.22) is $f_{uniform}$. The error between $f_{simulated}(t)$ and $f_{normal}(t)$ is defined as follows.

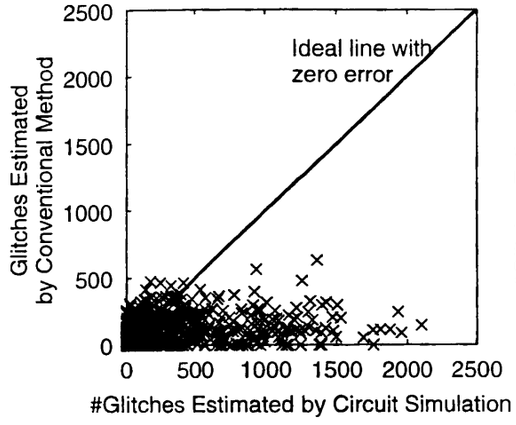
$$Error_{normal} = \int [f_{simulated}(t) - f_{normal}(t)]^2 dt. \tag{3.28}$$

$Error_{uniform}$ is also defined similarly.

$Error_{normal}$ and $Error_{uniform}$ are compared at all gates. In C3540 circuit, $Error_{normal}$ is smaller than $Error_{uniform}$ at the gates of 55%. On the other hand, $Error_{uniform}$ is smaller than $Error_{normal}$ at the gates of 55% in des circuit. Also the summations of $Error_{normal}$ and $Error_{uniform}$ for all gates are scarcely different, and the difference is within 1%. Even though the mean and the deviation are derived accurately, there is not a distinct difference in the error of the distribution function between $f_{normal}(t)$ and $f_{uniform}(t)$. The computational cost to construct $f_{uniform}(t)$ is much less than that of $f_{normal}(t)$. It can concluded that the uniform distribution is a reasonable and adequate shape for the use in glitch optimization.

3.4.2 Glitch Estimation

Now the accuracy of the proposed glitch estimation method is examined. The number of glitch transitions is estimated at every node in a circuit and it is compared to the value ob-



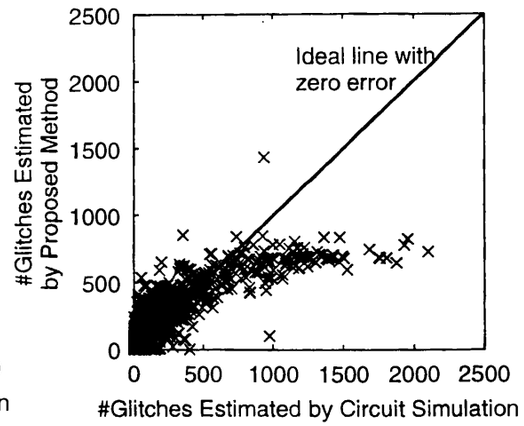


Figure 3.8: Accuracy Comparison of Glitches between Conventional and Proposed Method (i10).

tained by transistor-level simulation[41]. The glitch transitions are estimated in the following two ways.

Conventional Method: Only generated glitches are estimated (equivalent to [54] except for the simplified calculation of f function).

Proposed Method: Both generated and propagating glitches are estimated.

Fig. 3.8 shows the accuracy comparison of glitch estimation between the conventional method and the proposed method in i10 circuit. The horizontal axis represents the number of glitches estimated by transistor-level simulation. The vertical axis represents the number of glitches estimated by the conventional method or the proposed method. The correlation coefficient is calculated between simulated values and estimated values. The correlation coefficient of the proposed method is 0.84, whereas the coefficient of the conventional method is 0.38 in i10 circuit. The average correlation coefficients of the proposed method over 10 benchmark circuits are 0.74 and the coefficients of the conventional method is 0.38.

The accuracy of the estimated power dissipation is examined. The power dissipation is estimated using the proposed glitch estimation method that can consider propagating glitches and partial-swing transitions. Table 3.1 shows the result of power estimation. The column "Power" under "Simulation" represents the power dissipation evaluated by a transistor-level power simulator. The column "Error" under "Estimated" represents the estimation error of the proposed method for the power dissipation. The column "Time" represents the CPU time for power estimation on an Alpha Station. The average error of the power estimation is 13.8%. The CPU time required for the proposed method is more than 6000 times shorter

Table 3.1: Accuracy Comparison of Power Estimation between Conventional and Proposed Method.

Circuit	Simulation		Estimated			
	Power (mW)	Time (s)	Power (mW)	Error (%)	Time (s)	#gates
C3540	16.8	600	13.7	-18.5	0.02	766
ex5p	6.23	188	7.72	23.9	0.03	1041
misex3	13.2	442	14.2	7.6	0.03	1142
alu4	17.1	494	16.9	-1.2	0.03	1252
C5315	35.1	1115	28.0	-20.2	0.03	1334
i10	26.3	928	21.5	-18.3	0.03	1528
seq	14.8	520	15.5	4.7	0.04	1658
C7552	54.4	1483	41.9	-23.0	0.03	1670
des	43.1	1423	39.1	-9.3	0.06	2453
dsp_alu	193	11912	172	-10.9	0.25	6062
average†	-	-	-	13.8	-	-

average†: the average over the absolute amount of each error.

than that for a transistor-level power simulator, which enables to use the estimation method inside the optimization loop considering glitch reduction.

3.4.3 Optimization Algorithm

Next the effectiveness of the proposed optimization algorithm is examined. The proposed optimization algorithm is compared with a simple greedy algorithm and the simulated annealing method. The simple greedy algorithm calculates the sensitivity for all gates and resize a single gate with the largest sensitivity. After resizing the gate, the sensitivity of each gate is recomputed. If there are no gates which reduces the object function, the optimization loop finishes. The simple greedy algorithm is the same with the proposed algorithm in the case that *Max_Change*, *Reduce_Rate* and *Max_Iteration* are set to 1, 1.0 and ∞ respectively. The simulated annealing method is implemented as follows. A reconfiguration(move) is to select a gate randomly and resizing the gate to a size which is randomly decided. As for annealing schedule, temperature T is held constant during 100×(number of gates) reconfigurations or 10×(number of gates) successful reconfigurations. The temperature is decreased by the factor of 0.90. Table 3.2 shows the comparison of the optimization algorithms. The experiment is carried out using Eq.(3.8) as the object function. The column "Reduction" represents the percentage of the power reduction from the initial circuits. Here, in order to evaluate the optimization algorithm only, the power dissipation is estimated by the proposed glitch estimation method and Eq. (3.8). The column "Time" indicates CPU times for the optimization on an Alpha Station. In misex3 circuit, the greedy algorithm is trapped into a bad local solution and hence the reduction remains 8.8%, whereas the simulated annealing

Table 3.2: Comparison of Optimization Algorithms in Power Reduction and CPU Time.

Circuit	Greedy		Simulated Annealing		Proposed	
	Power Reduction (%)	Time (s)	Power Reduction (%)	Time (s)	Power Reduction (%)	Time (s)
C3540	6.0	36	6.6	6784	6.3	30
ex5p	19.7	336	25.1	14933	24.4	119
misex3	8.8	93	14.8	25037	15.2	101
alu4	5.9	93	12.2	13363	10.7	101
C5315	12.0	101	11.1	10914	12.2	63
i10	4.0	291	5.4	26682	5.8	189
seq	9.4	327	10.7	77027	12.6	277
C7552	6.6	94	6.3	21961	6.4	98
des	11.3	557	13.1	50878	12.4	413
dsp_alu	5.8	11584	7.4	1150867	5.7	7842
average	9.0	-	11.3	-	11.2	-

and proposed methods achieve more than 14 % reduction. The proposed algorithm reduces the power dissipation by 11.2% on average, whereas the greedy algorithm reduces by 9.0%. Also the CPU time spent for the proposed method is 79% of that for the greedy algorithm on average. Compared with the simulated annealing, the proposed algorithm can find a solution close to that of the simulated annealing, while spending only 0.6% of the CPU time on average.

3.4.4 Power Optimization

Here, the result of power optimization is shown. First, power dissipation is optimized without delay constraints. The given transition time constraint is 1.5ns. The initial circuits consist of the min-sized gates, except the gates up-sized for satisfying the transition time constraint, since the circuits are generated for minimizing area. The overall capacitive load of the initial circuits is almost minimum. Table 3.3 shows the result of the power optimization. The power dissipation before/after optimization is evaluated by a transistor-level power simulator. The column "Power(Delay) Reduction" represents the reduction of the power(delay) from the initial circuit. The column "Area Increase" shows the increase of the total cell area from the initial circuit. The proposed method increases the area by 5.2% on average. However the number of transitions are reduced by 8.5%, which mainly contributes to the power reduction of 10.4%. This means that the power dissipation of the circuits with the minimum active area is not minimum. It is notable that the delay is also reduced in all circuits, although the delay is not included in the objective function nor the constraints. The reduction of delay is 25.0% on average. Glitch reduction has an aspect of path balancing. The path balancing is enforced by reducing longer path delays, which leads to the reduction of the critical path delay.

Table 3.3: Power Optimization under No Delay Constraints.

Circuit	Power Reduction (%)	Delay Reduction (%)	Area Increase (%)	#Toggle Decrease (%)
C3540	5.9	9.1	2.9	5.5
ex5p	4.8	8.4	17.9	6.4
misex3	14.6	34.0	4.6	8.4
alu4	12.5	31.1	4.0	8.1
C5315	16.2	17.6	4.1	18.4
i10	8.6	30.8	6.0	5.7
seq	9.7	29.0	4.2	4.8
C7552	12.5	16.6	1.6	12.7
des	5.6	49.8	4.1	3.7
dsp_alu	14.0	23.2	2.2	11.0
average	10.4	25.0	5.2	8.5

Next the result of power optimization under delay constraints is presented and it is compared with the result with those of conventional methods. The circuit C5315 is optimized under a variety of delay constraints and the power dissipation is measured using a transistor-level power simulator. The circuit is optimized in the following three methods.

Delay Optimization: optimize delay only and do not care about power dissipation.

Conventional Method: optimize power dissipation based on the transition information of the initial circuit throughout the optimization process.

Proposed Method: optimize power dissipation by the proposed method.

The power-delay trade-off curve of each method is shown in Fig. 3.9. The initial circuit is located near the top right corner of the figure. Achievable delay times by the three methods are the same. The fastest circuits by the three methods have 5.9ns delay time. However the power dissipation is different and, as expected, the proposed method provides the lowest. Because the reduction of the delay time and path balancing lie in the same direction, it is seen that delay reduction does not increase power dissipation so much. Indeed, the fastest circuit obtained by the delay optimization method has the total cell area 14 % larger than that of the initial circuit, while the power dissipation is almost the same as that of the initial circuit. Corresponding increase in capacitive load is compensated by the reduction of glitch activity which is a by-product of the delay optimization. The conventional method which assumes constant glitch activities throughout the optimization process does not work well, compared with the proposed method. It is because the glitch activities are changing in the optimization process. In order to reach good solutions, the fact that glitches are affected by gate resizing has to be considered. Explicitly exploiting the possibility of glitch reduction,

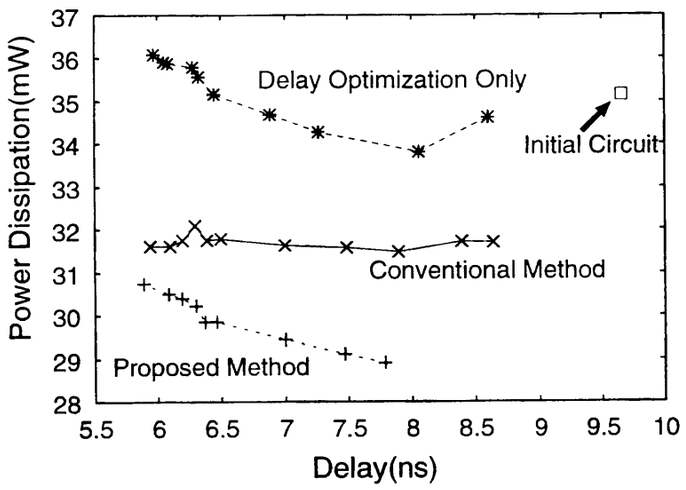


Figure 3.9: Power-Delay Trade-Off Curve (C5315).

the proposed method further reduces the power dissipation. It can be seen that the gate sizing considering glitch reduction is an effective method for power reduction.

3.4.5 Tolerance to Skew Fluctuation and Wire Capacitance Variation

In actual circuits, there are various factors that change delay characteristics, such as skew fluctuations, variations in transistor characteristics and wire capacitances. The tolerance of the proposed method to uncertainties in delay characteristics is examined.

First, power dissipation is examined under skew fluctuations. The skew time at each primary input is assumed to fluctuates according to the normal distribution(0, σ). Power dissipation is optimized by the following two methods.

Sizing(A): optimization that does not consider skew fluctuations, i.e. only the first term in Eq. (3.27) is considered.

Sizing(B): optimization that considers skew fluctuations(Sec. 3.2.6).

100 sets of skew patterns are generated for 3σ of skew fluctuation being 0.5ns and 1.0ns. In this fabrication process, the delay time of a single inverter with fanout loading three is 0.1ns. The number of applied pattern for power evaluation is 100 because of the enormous simulation cost. Fig. 3.10 shows the relationship between the amount of power reduction and skew fluctuations. It can be seen that the proposed method can reduce power dissipation under skew fluctuation. Owing to the consideration of the skew fluctuation, the average value of the power reduction becomes about 1% larger. In the case of $3\sigma = 0.5$ns, Sizing(B) is much effective than Sizing(A). The reason is guessed such that the consideration for skew

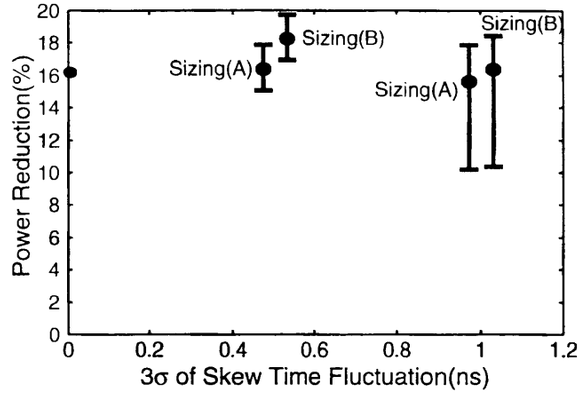


Figure 3.10: Power Reduction under Skew Fluctuations (C5315).
Sizing(A): optimization that does not consider skew fluctuations.
Sizing(B): optimization that consider skew fluctuations.

fluctuation compensates not only skew fluctuations but also the error in delay calculation as a by-product.

Because of manufacturing variability, wire capacitance fluctuates. Also wire load estimation contains a certain amount of error. Therefore the gate delay has some amount of uncertainty. Power dissipation is evaluated under wire load fluctuations. Wire capacitance is assumed to fluctuate according to the normal distribution(0, σ). 100 sets of wire load are generated and power dissipation is evaluated using them. The ratio of total gate capacitance and the total wire capacitance is about 1:2 in this circuit. The relationship between power reduction and the amount of wire capacitance fluctuations is shown in Fig.3.11. The average reduction at each 3σ value is almost the same. Even in the worst case of $3\sigma = 40\%$, the power dissipation is reduced by 14.8%. It can be seen that the proposed method is effective under uncertainties in delay characteristics that exist in fabricated circuits.

3.5 Conclusion

This chapter proposes a power optimization method by gate sizing. The proposed method optimizes not only the amount of capacitive load and short-circuit current but also the number of glitch transitions. A statistical glitch estimation method, which can consider propagating glitches, partial-swing transitions and skew fluctuation, is devised. The proposed gate resizing algorithm has both the merit of rapid convergence and the ability to get out of a bad local solution. The effect of the proposed method is experimentally verified using 10 benchmark circuits with a 0.6 μm standard cell library. The power dissipation is reduced from the minimum-area circuits by 10.4 % on average and by 16.2 % maximum. It is observed that the conventional method, which assumes that glitches do not change by gate resizing, does not achieve sufficient power reduction. On the other hand, the proposed method can

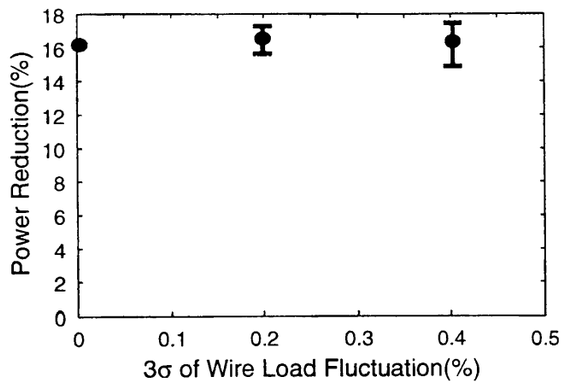


Figure 3.11: Power Reduction under Wire Capacitance Fluctuations (C5315).

reduce power dissipation further guided by the proposed glitch estimation method. It is also verified that the proposed method is effective under manufacturing variability such as skew time fluctuation and wire capacitance variation.

Chapter 4

Performance Optimization by Gate Sizing Based on Statistical Static Timing Analysis

This chapter discusses a gate resizing method for performance enhancement based on statistical static timing analysis. The proposed method focuses on timing uncertainties caused by local random fluctuation. The proposed method aims to remove both over-design and under-design of a circuit, and realize high-performance and high-reliability LSI design. The effectiveness of the proposed method is examined by 6 benchmark circuits. The experimental results show that the proposed method can reduce the delay time further from the circuits optimized for minimizing the delay without the consideration of delay fluctuation.

4.1 Introduction

There are several sources that cause the uncertainties of circuit delay time, such as manufacturing fluctuation, estimation error of wire capacitance and resistance, uncertainties of wire capacitance during physical design, supply voltage and temperature change, diversity in signal waveforms, and so on. These sources can be classified into two categories. The first category is a global change that applies to all gates and wires similarly in a certain region. The second category is a random change that indicates a certain statistical distribution. As for the global change, there is a traditional and widely-used method to consider the delay time uncertainties. In this method, three values(best/typical/worst-case values) are prepared for the delay time of each gate and wire. Then the circuit delay time is calculated using each-case value for purpose by purpose. This is a reasonable approach for the global change.

On the other hand, the random change is not well considered in LSI design. Due to the random change, the delay time of each gate and wire has a certain probability distribution. In one case, a certain amount of design margin is set to avoid the effect of the delay time uncertainties by the random change. In this method, the decision of the design margin is difficult, which results in excessive design margin and over-design of the circuits. In another

case, the delay time of each gate and wire is defined as the worst-case value, for example, mean+3σ. In this case, the estimated delay time of a critical path is pessimistic, and the delay of the shortest path can not be considered. Therefore, in order to design a circuit with high confidence and eliminate over-design, a statistical static timing analysis method and a circuit optimization method considering the random change are necessary.

This chapter proposes a performance optimization method considering the random change based on statistical timing analysis. As for statistical timing analysis, there are several proposals [57, 58, 59, 55, 60]. The methods proposed in Refs. [57, 58, 59] are Monte Carlo simulation-based techniques, so these methods are not suitable for performance optimization method from the point of computation time. The method proposed by Berkelaar in Ref. [55, 60] is based on a static timing analysis method. This method does not require any simulations, and the complexity of the timing analysis is linear to the circuit scale. So the timing analysis can be done in a realistic computation time. Although this method works well for the estimation of the mean delay, it underestimates the worst delay(corresponding to mean+3σ, for example)[55], because of the definition of the worst-case delay and the approximation method used in Ref. [55]. In a statistical analysis, it is important to estimate a statistically well-defined worst-case value. Therefore the worst-case delay is defined in a statistical manner, and a technique to improve the accuracy of the worst-case delay estimation is devised. This method is utilized for performance optimization.

In the case of the performance optimization based on statistical static timing analysis, slack[40], which represents the timing criticality at each gate and is widely used for performance optimization under deterministic delay model, can no longer be a useful measure under statistical environment. This chapter therefore proposes a new measure "criticality" that represents the timing criticality at each gate, and device performance optimization algorithms utilizing the "criticality". In Ref. [60], the gate sizing problem is formulated as a nonlinear programming problem, where the objective function and the constrains are expressed as analytic forms. In this method, the delay should be represented by a simple analytical equation, which degrades the accuracy of the delay calculation. On the other hand, the proposed method can utilize any gate/wire delay calculation methods.

The proposed performance optimization method has various applications, such as uncertainties of wire capacitance during physical design, local fluctuation in transistor characteristics, local variation of supply voltage and temperature, and so on. The proposed performance optimization method can eliminate over-design of a circuit and contribute high-performance and high-reliability LSI design.

This chapter is organized as follows. Section 4.2 discusses the statistical static timing analysis method. Section 4.3 explains the performance optimization algorithms of gate sizing. Section 4.4 discusses some applications of the proposed performance optimization method. Section 4.5 demonstrates some experimental results. Finally, Section 4.6 concludes the discussion.

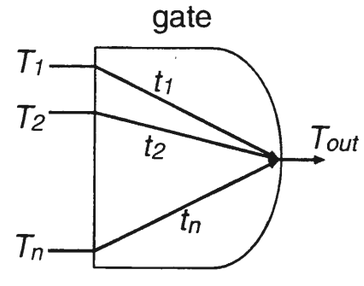


Figure 4.1: Gate Delay Model.

4.2 Statistical Static Timing Analysis

In this section, a statistical timing analysis method is discussed. First, the basic concept of the statistical static timing analysis proposed in Ref. [55] is explained. Next, approximation methods of the delay distribution used in the statistical static timing analysis are discussed. This chapter then proposes a new measure "criticality" that represents the timing criticality at each gate.

4.2.1 Static Timing Analysis

First a conventional(not statistical) static timing analysis method is explained briefly. Suppose a gate that has n-input and 1-output ports(Fig. 4.1). T_i is the latest arrival time of signals at the i-th input. t_i is the gate delay time from the i-th input to the output. T_i and t_i have different values for rise and fall transitions. In Section 4.2, rise/fall transitions are not distinguished for simplifying the explanation. But the real implementation in Section 4.5 considers the delay difference for rise/fall transitions. The latest arrival time of the signal transitions at the output, T_{out}, is represented as follows.

$$T_{out} = \max_{i=1}^{n}(T_i + t_i). \tag{4.1}$$

Using Eq. (4.1), the latest arrival time at each gate can be calculated incrementally without tracing all paths.

4.2.2 Statistical Static Timing Analysis

In a conventional static timing analysis, each delay time of gates and wires is a constant value. On the other hand, under the existence of uncertainties in circuit delay time, each delay time is not a constant and it has a statistical distribution, which is considered for delay calculation in the statistical static timing analysis. The basic concept of the statistical static timing analysis has been proposed in Ref. [55]. This method is explained briefly. Next,

the worst-case delay of the circuit with delay fluctuation is defined, and a technique that improves the accuracy of the worst-case delay calculation is discussed.

The distribution of the latest signal arrival time at the i-th input is modeled as a normal distribution of a stochastic variable T with mean μ_{T_i} and standard deviation σ_{T_i}. It is also assumed that the gate delay time from the i-th input to the output is distributed normally with a stochastic variable t, mean μ_{t_i} and standard deviation σ_{t_i}.

Here, Eq. (4.1) is converted for the statistical timing analysis. The probability density function f_i is defined such that f_i expresses the distribution of T_i+t_i. The distribution of f_i becomes a normal distribution $\mathrm{N}(\mu_{T_i}+\mu_{t_i}, \sqrt{\sigma_{T_i}^2+\sigma_{t_i}^2})$. The cumulative distribution function F_i is defined as follows.

$$F_i(x) = \int_{-\infty}^{x} f_i(\chi)d\chi. \tag{4.2}$$

As an example of statistical max operation, $C = \max(A, B)$, with stochastic variables A, B and C, is examined. In this case, the following relation holds at any x.

$$P(C \leq x) = P((A \leq x) \cap (B \leq x)), \tag{4.3}$$

where $P(Condition)$ represents the probability that *Condition* is satisfied. When the statistical correlation between A and B is ignored, Eq. (4.3) can be transformed as follows.

$$P(C \leq x) = P(A \leq x) \cdot P(B \leq x). \tag{4.4}$$

The probability density functions of A, B and C are defined as f_A, f_B and f_C. Eq. (4.4) can be expressed as follows.

$$\int_{-\infty}^{x} f_C d\chi = \int_{-\infty}^{x} f_A d\chi \cdot \int_{-\infty}^{x} f_B d\chi. \tag{4.5}$$

Differentiating Eq. (4.5), the following equation can be obtained.

$$f_C(x) = f_A(x) \cdot \int_{-\infty}^{x} f_B d\chi + f_B(x) \cdot \int_{-\infty}^{x} f_A d\chi. \tag{4.6}$$

Eq. (4.6) can be rewritten as follows.

$$P(C = x) = P(A = x) \cdot P(B \leq x) + P(B = x) \cdot P(A \leq x). \tag{4.7}$$

Extending Eq. (4.6) for n stochastic variables, the probability density function f_{out}, which corresponds to the distribution of the latest arrival time T_{out}, can be represented as follows.

$$f_{out}(x) = \sum_{i}^{n} \left[f_i(x) \cdot \prod_{j \neq i}^{n} F_j(x) \right]. \tag{4.8}$$

The probability density function of the overall circuit delay time can be obtained by applying the probability density function at each primary output to f_i.

The definition of the worst-case delay under the statistical delay model is discussed. The distribution of the latest arrival time, f_{out}, is different from a normal distribution, though

assumed to be normal. Figs. 4.2 and 4.3 show an example of the difference between f_{out} and the normal distribution. The function f_{out} represents Eq. (4.8) under the following conditions. The mean and standard deviation of f_1, the mean and standard deviation of f_2 and n are 3, 1, 3.6, 0.6 and 2 respectively. The mean m and standard deviation σ of f_{out} are calculated according to the definition, and the normal distribution $\mathrm{N}(m, \sigma)$ is generated. If the distribution of f_{out} is exactly normal, x_1 in the following equation becomes equal to $m + 3\sigma$.

$$0.9986501 = \int_{-\infty}^{x_1} f_{out}(x)dx, \tag{4.9}$$

where the value 0.9986501 is the probability of a normal distribution between $-\infty$ and $m + 3\sigma$. But in reality, x_1 of f_{out} is different from $m + 3\sigma$. The value x_1 is 6.00, whereas $m + 3\sigma$ is 5.64. This difference derives from the fact that the curve of f_{out} falls slower than it rises. If the worst-case delay is defined as $m + 3\sigma$, the lower probability of $x \leq m + 3\sigma$ becomes smaller than 99.87%. The actual value of the lower probability varies depending on the shapes of f_1 and f_2. On the other hand, when the worst-case delay is defined as x_1, the lower probability of $x \leq x_1$ becomes a fixed value of 99.87%. In statistical analysis, evaluating the delay time with a fixed lower probability is important. Therefore the worst-case delay is defined as x_1 in Eq. (4.9). When the delay with the different lower probability is evaluated, the value of the left term in Eq. (4.9) should be changed accordingly. Hereafter, the worst-case delay is defined as x_1 in Eq. (4.9).

Next, the approximation of f_{out} to a normal distribution is discussed. In Ref. [55], f_{out} is approximated as a normal distribution to reduce computational costs. The proposed method also approximate f_{out} to a normal distribution. Here, the approximation methods of f_{out} are examined from the viewpoint whether the worst-case delay x_1 can be calculated accurately. Eq. (4.9) is rewritten using Eq. (4.8) as follows.

$$0.0013499 = \int_{x_1}^{\infty} \sum_{i}^{n} \left[f_i(x) \cdot \prod_{j \neq i}^{n} F_j(x) \right] dx. \tag{4.10}$$

The value x_1 of each f_i is close to or smaller than x_1 of f_{out}. In the range of x between x_1 and ∞, the cumulative distribution function $F_j(x)$ is almost 1. In order to calculate the worst-case delay x_1 accurately, the approximation accuracy of f_i where x is larger than x_1 is important. Therefore f_{out} should be approximated well in the region where x is close to and larger than x_1 of f_{out}, which contributes the accurate calculation of x_1 at the fan-out gates that the gate drives. Two approximation methods of f_{out} to a normal distribution $\mathrm{N}(m, \sigma)$ are compared.

Method 1 Calculate the mean m and the standard deviation σ of f_{out} according to the definition.

Method 2 Find the values of x_0 and x_1 that satisfy Eqs. (4.9) and (4.11). The mean m is calculated as $(x_0 + x_1)/2$ and the standard deviation σ is $(x_1 - x_0)/6$.

$$0.0013499 = \int_{-\infty}^{x_0} f_{out}(x)dx. \tag{4.11}$$

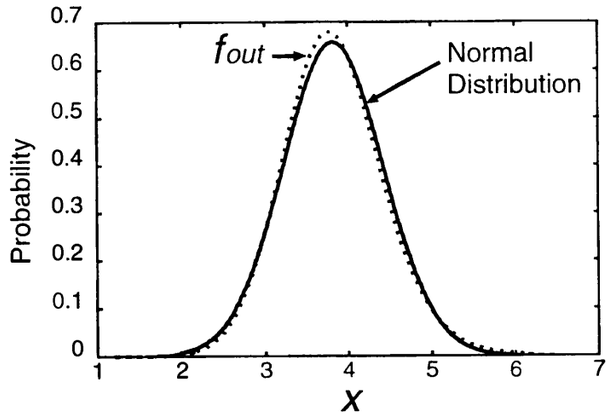


Figure 4.2: Difference between f_{out} and a Normal Distribution.

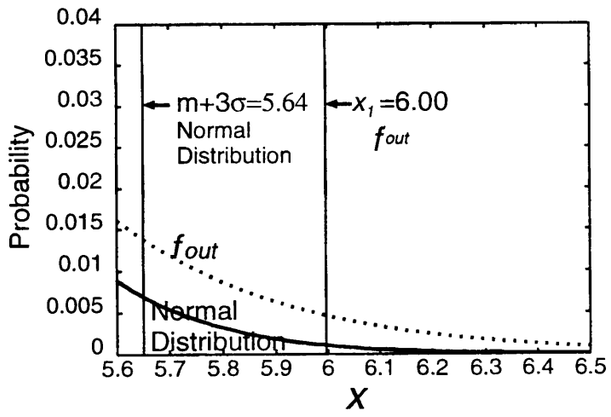


Figure 4.3: Difference between f_{out} and a Normal Distribution(Magnified).

Method 1 is adopted in Ref. [55]. In Method 2, a value x_0 corresponds to $m - 3\sigma$ of a normal distribution and x_1 to $m + 3\sigma$ from the viewpoint of the lower and upper probability. Method 2 adjusts x_1 of the approximated normal distribution to x_1 of f_{out}. Fig. 4.4 shows the approximation results of Method 1 and Method 2. Method 1 underestimates the delay time. On the other hand, in Method 2, the distribution shape of f_{out} where x is larger than x_1 is well approximated. Therefore, Method 2 is suitable for the approximation to calculate the worst-case delay x_1 accurately. When the definition of the worst-case delay is changed, i.e. the value of the left term in Eq. (4.9) becomes other value, Method 2 is modified as follows. For

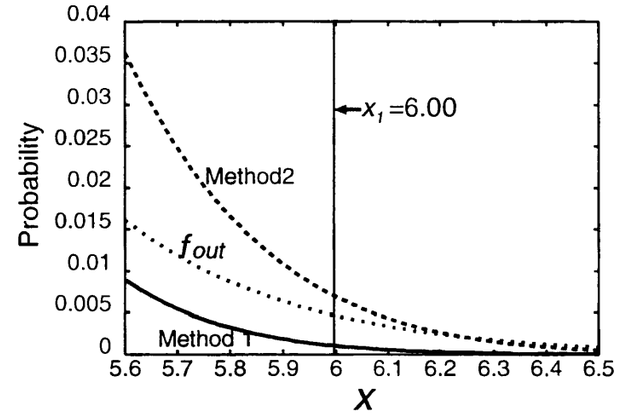


Figure 4.4: Approximation to Normal Distribution(Magnified).

example, suppose the value of the left term in Eq. (4.9) becomes 0.97725, which corresponds to the probability of $x \leq m+2\sigma$ in a normal distribution. The value of left term in Eq. (4.11) becomes 0.02275. The standard deviation σ is calculates as $(x_1 - x_0)/4$.

The discussion so far assumes that the distribution of gate delay is normal and hence the probability density function f_i is a normal distribution. In this case, the probability density function f_{out}, although it is not a normal distribution, can be approximated to a normal distribution. Two methods for the approximation are shown. Please notice that the essence of the statistical static timing analysis explained from Eqs. (4.2) through (4.10) does not require that f_i is normal. Thus, if the probability density function f_i is not normal, Eq. (4.8) can be still applied to calculate the probability density function f_{out}. In this case, another appropriate function for f_i and f_{out} is needed, or numerical calculation of f_i and f_{out} is required. In any case, through the successive calculation of the probability density function from the primary input to the primary output, statistical static timing analysis can be performed.

4.2.3 Criticality

In the case of a conventional(not statistical) static timing analysis method, slack is a useful measure that represents the timing criticality at each gate[40]. Many performance optimization algorithms using slack have been proposed[61, 62, 45], and slack helps to reduce the computation time required for the optimization considerably. But in the statistical static timing analysis, slack can not be used as a measure of timing criticality. Since slack is defined as the time difference between the required arrival time and the latest arrival time, the required arrival time at each gate is computed from the primary outputs. In statistical static timing analysis, the required arrival time at each input can not be calculated independent of the arrival times at the other inputs. It is because the arrival time at the output is affected by

all the inputs' arrival time(Eq. (4.8)). Thus, the required arrival time can not be propagated. Also the combination of the mean m and the standard deviation σ at each gate, which satisfies the delay constraint, is not determined uniquely. So, the required arrival time can not be defined. This chapter therefore introduces a new measure "criticality" that represents the timing criticality at each gate.

Before the detailed explanation of "criticality", the concept of "criticality" is explained. Under the statistical delay model, many paths have a possibility to become the longest path. In other words, many gates have an effect to the distribution of the total circuit delay. To speak more rigidly, all gates have an influence to the circuit delay distribution although the magnitude of the influences is different. Therefore, the timing criticality at each gate should be defined as the magnitude of the statistical influence to the circuit delay distribution. Namely, the gate that has a strong statistical influence to the total delay distribution should be defined as critical. The statistical impact of each gate delay to the total circuit delay is modeled as the measure of timing criticality named "criticality", using a heuristic numerical expression. In this model, large "criticality" represents high timing criticality, thus the gate with large "criticality" should be resized for reducing the circuit delay. When "criticality" is zero, the gate has no statistical influence to the circuit delay distribution. So, the gate with small "criticality" could be downsized for reducing power dissipation without delay increase. Given the measure of "criticality", the proposed method can choose a candidate of gate resizing efficiently. Hereafter, the details of "criticality" is explained.

The term in the bracket of Eq. (4.8) represents the following probability.

$$f_i(x) \cdot \prod_{j \neq i}^{n} F_j(x) = P(T_i + t_i = x) \cdot \prod_{j \neq i}^{n} P(T_j + t_j \leq x). \tag{4.12}$$

The input with the high probability of Eq. (4.12) affects the distribution of T_{out} at x strongly. The probability of Eq. (4.12) expresses the magnitude of the influence that the i-th input gives to f_{out} at x. "$influence_i$" is defined such that it represents the influence proportion of the i-th input in the range of $x \geq x_1$ as follows.

$$influence_i = C_1 \cdot \int_{x_1}^{\infty} f_i(x) \cdot \prod_{j \neq i}^{n} F_j(x) \cdot \exp(C_2 \cdot x) dx, \tag{4.13}$$

where C_1 is a normalization coefficient to satisfy $\sum_i^n influence_i = 1$ and C_2 is a constant. A term $exp(C_2 \cdot x)$ is multiplied in order to emphasize the region of large arrival time. However, this is not a primary term for the definition of $influence_i$. Also, according to the experiments, the value of C_2 is not so sensitive to $influence_i$. The value of C_2 is empirically decided such that the value $exp(C_2 \cdot x)$ increases by 50% when time x increases by 0.1ns around the time of interest. When $influence_i$ is 1, f_{out} in $x \geq x_1$ is determined by the i-th input and the other inputs do not affect f_{out}. Conversely, when $influence_i$ is 0, the i-th input does not influence on f_{out} in $x \geq x_1$ at all. "Influence" at each primary output on the overall circuit delay time can be similarly obtained by applying the probability density function at each primary output to f_i.

Now the calculation method of "criticality" that represents the timing criticality at each gate is explained. "Criticality" at each gate is defined as the amount of the contribution to the

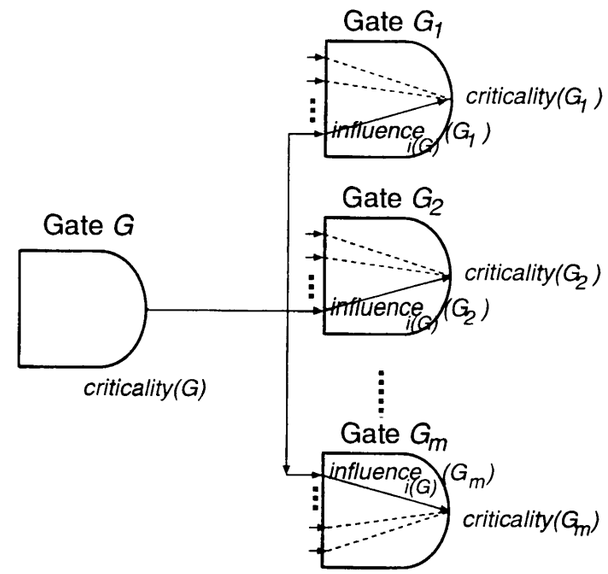


Figure 4.5: Propagation of "Criticality".

circuit delay by the paths that go through the gate. "criticality" is propagated from primary outputs to primary inputs. Suppose Fig. 4.5 given for an example. $i(G)$ is defined such that the $i(G)$-th input is connected with gate G. A term $influence_{i(G)}(G_j)$ means how much the $i(G)$-th input affects the timing at gate G_j in $x \geq x_1$. In other words, $influence_{i(G)}(G_j)$ represents how easily the timing criticality propagates from gate G_j to gate G. Therefore "criticality" propagated from gate G_j to gate G is represented as $influence_{i(G)}(G_j) \cdot criticality(G_j)$.

$$criticality(G) = \sum_{j}^{m} influence_{i(G)}(G_j) \cdot criticality(G_j), \tag{4.14}$$

where m is the number of fan-outs for gate G. At primary outputs, "influence" means the timing criticality itself. It is because the primary output with large "influence" affects the circuit delay strongly, i.e. the timing criticality is high. So, "criticality" at primary outputs is set to 1, which enables that Eq. (4.14) is hold even when G_j is a primary output. "criticality" can be calculated by the breadth-first trace from the primary outputs.

The complexity of this statistical timing analysis method and the calculation of "criticality" is linear to the circuit scale. This property of the complexity make it possible to estimate and optimize the circuit delay of a large circuit.

4.3 Optimization Algorithm

This section explains a performance optimization algorithm based on statistical static timing analysis by gate resizing. Two algorithms are shown, one is for delay optimization and the other is for power(area) optimization. These algorithms utilizes "criticality" explained in the previous section.

4.3.1 Delay Optimization

The delay optimization algorithm is shown below.

Step 1: put all gates into list L.
Step 2: if L is empty or delay constraint is satisfied, finish optimization.
Step 3: find the gate with maximum criticality in L.
Step 4: resize the gate to the size with minimum delay.
Step 5: if there are no sizes to reduce delay, remove the gate from list L and go back to Step 2.
Step 6: go back to Step 1.

First all gates are put into the list L of the resizing candidate. When the candidate list L is empty or the delay constraint is satisfied, the optimization process finishes. The gate with maximum criticality in L is searched. It is because the gate with large criticality affects the circuit delay time strongly. The size of the gate is changed four times, i.e. 2 size-up, 1 size-up, 1 size-down, and 2 size-down, and evaluate the circuit delay for each case. The size that the circuit delay decrease the most is chosen, and the gate is resized to the size. If the resizing does not decrease the circuit delay, the gate size is restored and the gate is removed from L, and the optimization process goes back to Step 2. Otherwise, the optimization process goes back to Step 1. The proposed algorithm searches a solution greedily, so the proposed algorithm necessarily reaches the condition that the circuit delay does not decrease by resizing the gates in the circuit. In this condition, as the steps between Step 2 and Step 5 are repeated, the number of the elements in the list L decreases. Finally the list L becomes empty and the optimization procedure finishes in Step 2.

4.3.2 Power(Area) Optimization under Delay Constraint

The gate resizing algorithm for power(area) reduction is explained.

Step 1: put all gates into list L.
Step 2: if L is empty, finish optimization.
Step 3: find the gate with minimum criticality in L.
Step 4: resize the gate to the size with minimum power dissipation without delay violation.
Step 5: if there are no gate sizes to choose, remove the gate from list L and go back to Step 2.
Step 6: go back to Step 1.

First all gates are put into the list L of the resizing candidate. When the candidate list L is empty, the optimization process finishes. The gate with minimum criticality in L is searched, because the gate with small criticality scarcely influences on the circuit delay. The size of the found gate is changed to 2 size-down and 1 size-down from the initial size, and the circuit delay and the power dissipation are evaluated for each case. The found gate is down-sized to the size that makes the power dissipation minimum without the delay violation. If the resizing does not reduce power dissipation without delay violation, the gate is removed from L and the optimization process goes back to Step 2. Otherwise, the optimization process goes back to Step 1. At the end of the optimization, there become no gates to reduce power dissipation without delay violation. The list L becomes empty by the repetitions between Step 2 and Step 5, and finally the optimization procedure finishes.

The optimization algorithm explained above has the possibility of falling into a bad local minimum solution. In order to escape from a bad local minimum solution, the circuit delay is optimized a little bit, such as 0.1% of its circuit delay, using the algorithm in Sec. 4.3.1. After that, the above algorithm is applied again. This loop is repeated for several times.

4.4 Applications

This section shows some applications of the statistical timing analysis method and the optimization algorithm explained in previous sections. Performance optimization based on the statistical timing analysis has a considerable possibility to contribute high-performance and high-reliability LSI design. It is assumed that the gate delay fluctuation discussed in this section can be approximated to a normal distribution. If the distribution is not normal, statistical timing analysis can be still performed as described in Sec. 4.2.2. In this case, the modification of the method for expressing the probability density functions is needed.

4.4.1 Uncertainties of Wire Capacitance during Physical Design and Uncertainties in Signal Waveforms

As the influence of wire on the circuit delay increases, timing closure has become a serious problem. This problem is caused by the uncertainties of wire capacitance during physical design. Also, the wire capacitance estimated from a final layout has a certain amount of errors. Because of the simple definition of the transition time, there are many different waveforms that have the same transition time, which causes the gate delay uncertainty. When the gate delay is derived from the two-dimensional look-up table with capacitive load and transition time as parameters, the gate delay is represented as follows.

$$delay = a_0 + a_1 \cdot t_{tran} + a_2 \cdot c_{load} + a_3 \cdot t_{tran} \cdot c_{load}, \tag{4.15}$$

where a_0, a_1, a_2 and a_3 are the constants decided by the look-up table, c_{load} is the load capacitance and t_{tran} is the transition time of the input signal. If the uncertainties of c_{load} at each design phase and t_{tran} can be modeled properly, the distribution of the gate delay can be

derived. Then, the proposed performance optimization method can eliminate the excessive design iteration and the over-design.

4.4.2 Local Fluctuations in Transistor Characteristics, Supply Voltage and Temperature

The local variation of the transistor characteristics is represented as the fluctuation of the device parameters(v_t, β, ...) and the process parameters(t_{ox}, W, L, ...). The operating parameters(V_{DD}, $Temp$) also fluctuate locally. The gate delay time $delay$ can be represented as a function of p_i, where p_i corresponds to each device, process, or operating parameters. When the local changes are not so large, the change of the gate delay time $\delta delay$ can be represented as follows.

$$\delta delay = \sum_i d_i \cdot \delta p_i, \tag{4.16}$$

where d_i is a constant. In the case of the local fluctuation, δp_i varies according to a certain statistical distribution. The distribution of the gate delay time can be obtained. With the derived delay distribution, the circuits can be optimized considering the local fluctuations.

4.5 Experimental Results

In this section, some experimental results are shown. First the accuracy of the worst-case delay estimation is verified. The next experiment demonstrates the delay fluctuation caused by the timing uncertainties of local random change. Finally the delay and power optimization results under the condition that the wire capacitance fluctuates are shown.

The circuits used for the experiments are taken from ISCAS85 and LGSynth93 benchmark sets. These circuits are synthesized and mapped by a commercial logic synthesis tool[56] under a reasonable wire load model such that the power dissipation is minimized under the following four delay constraints. The circuits labeled "_A" are generated under the minimum as well as reachable delay constraints of the respective circuits. The delay constraints given to the circuit with "_B", "_C" and "_D" are made loose gradually in this order. The ratio of the total gate capacitance and the total wire capacitance is about 1:1. The target library is a standard cell library used for actual fabrication in a 0.35 μm process with three metal layers. The library includes basic and complex gates. Buffer and Inverter have eleven varieties in the driving strength and other gates have six varieties. A typical delay time at each gate is calculated based on two-dimensional look-up tables with capacitive load and slew as parameters. The delay difference between rise/fall transitions is considered. The energy dissipated at each gate, which includes capacitive and short-circuit power dissipation, is derived from a look-up table with capacitive load and slew as parameters. The look-up tables of the gate delay, the transition time of the output signal and the power dissipation are characterized by circuit simulation. As for the power evaluation, it is assumed that all gates have the same switching probability of 0.2 and the cycle time of the input patterns is 100ns.

4.5.1 Accuracy of Worst-Case Delay Calculation

The accuracy of the worst-case delay calculation is verified. Each gate delay time is assumed to fluctuate according to normal distribution. The mean is the typical gate delay time and the standard deviation is 20% of its gate delay time. The worst-case delay time defined as x_1 in Eq. (4.9) is evaluated. Three methods are compared, Monte Carlo simulation, the statistical static timing analysis with Method1(Section 2.2) which is equivalent to Ref. [55], and the proposed statistical static timing analysis with Method2(Section 2.2). In Monte Carlo simulation, the number of evaluation is 100,000. The comparison of the accuracy is shown in Table 4.1. The column under "Typ. Delay" is the circuit delay time with no delay fluctuation. The columns "Monte Carlo", "SSTA[55]", "Proposed SSTA" correspond to the results of Monte Carlo simulation, the statistical static timing analysis in Ref. [55] and the proposed statistical static timing analysis respectively. The columns "Delay" are the worst-case delay time of the circuits with delay fluctuation. "Increase" means the proportion of the difference between the typical(no fluctuation) delay and the worst-case delay with delay fluctuation. "Error" represents the estimation error compared with Monte Carlo simulation. The range of the estimation error in the proposed method is $-0.8 \sim 2.9\%$, and the average error is 1.4%. As for SSTA[55], the range is $-6.7 \sim -2.7\%$, and the average is 4.3%. The improvement of the approximation to normal distribution contributes a better calculation of the worst-case delay x_1.

4.5.2 Circuit Delay Fluctuation – Case Study –

The circuit delay fluctuation caused by the timing uncertainties of local random fluctuation is demonstrated. First the delay uncertainty sources are discussed, and an assumption of the delay uncertainty sources is made. Then the result of the statistical static timing analysis under this assumption is shown.

Assumption of Delay Fluctuation Sources

As for the sources of delay fluctuation, two sources are considered; manufacturing variability and design uncertainties of wire capacitance.

Manufacturing Variability

The manufacturing variability consists of two components; the variability in transistor characteristics and the variability in interconnect structure. First the transistor characteristics is discussed. The fluctuation is composed of local components(different for individual gates in a circuit) and global components(the same for all gates in a circuit)[25]. In the process used for the experiments, the worst-case delay evaluated from the given worst-case SPICE parameters is 30% larger than the typical-case delay. Thus, if the ratio of the local fluctuation component and the global fluctuation component is assumed to be 2:1, 3σ of the local delay variability becomes 20%.

Table 4.1: Accuracy of Worst-Case Delay Calculation.

Circuit	Typ. Delay (ns)	Monte Carlo Delay (ns)	Monte Carlo Increase (%)	SSTA[55] Delay (ns)	SSTA[55] Error (%)	Proposed SSTA Delay (ns)	Proposed SSTA Error (%)
C432_A	4.48	5.57	24.3	5.39	-3.2	5.65	1.3
C432_B	4.97	6.10	22.7	5.90	-3.3	6.19	1.5
C432_C	5.91	7.13	20.6	6.94	-2.7	7.26	1.8
C432_D	6.92	8.58	24.0	8.35	-2.7	8.79	2.4
C3540_A	6.71	8.28	23.4	7.97	-3.7	8.43	1.8
C3540_B	7.18	8.77	22.1	8.45	-3.6	8.95	2.1
C3540_C	7.97	9.65	21.1	9.30	-3.6	9.80	1.6
C3540_D	8.92	10.69	19.8	10.32	-3.5	10.90	2.0
C5315_A	6.00	7.73	28.8	7.31	-5.4	7.83	1.3
C5315_B	6.97	8.58	23.1	8.26	-3.7	8.74	1.9
C5315_C	7.98	9.74	22.1	9.48	-2.7	10.02	2.9
C5315_D	8.90	10.77	21.0	10.47	-2.8	11.03	2.4
C7552_A	4.84	6.12	26.4	5.86	-4.2	6.20	1.3
C7552_B	5.02	6.28	25.1	5.98	-4.8	6.33	0.8
C7552_C	5.99	7.39	23.4	7.07	-4.3	7.48	1.2
C7552_D	6.95	8.53	22.7	8.18	-4.1	8.68	1.8
alu4_A	3.31	4.25	28.4	4.00	-5.9	4.23	-0.5
alu4_B	3.99	5.10	27.8	4.76	-6.7	5.10	0.0
alu4_C	4.95	6.18	24.8	5.82	-5.8	6.14	-0.6
alu4_D	5.83	7.26	24.5	6.80	-6.3	7.20	-0.8
des_A	3.60	4.73	31.4	4.52	-4.4	4.78	1.1
des_B	3.98	5.26	32.2	5.00	-4.9	5.26	0.0
des_C	4.96	6.50	31.0	6.12	-5.8	6.46	-0.6
des_D	5.91	7.52	27.2	7.17	-4.7	7.59	0.9
	-	-	24.9	-	4.3	-	1.4

Next the variability in interconnect structure is examined. Reference[63] analyzes the decomposition of the delay variability due to manufacturing fluctuation. The analysis indicates that the interconnect is responsible for 12 to 18% of the total delay variability and the rest (82 to 88%) is contributed by transistors. With this ratio of each contribution, 3σ of the total delay variability becomes 24%. Thus, in this case study, the standard deviation of the delay due to transistor and interconnect variabilities is estimated to be 8%.

Design Uncertainties of Wire Capacitance

The estimated wire capacitances during layout design are different from the capacitances of the final layout. At cell placement design phase, there are uncertainties in wire route and

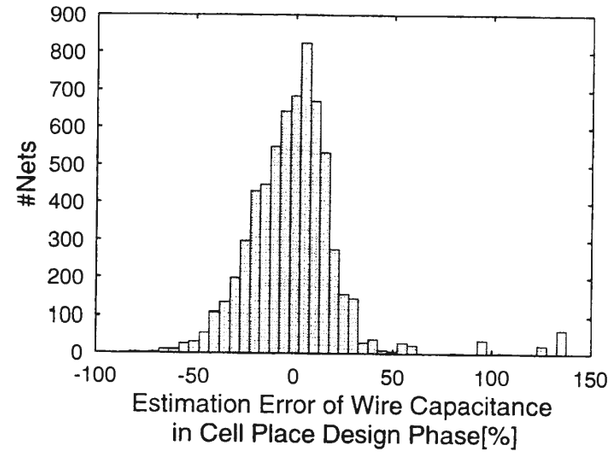


Figure 4.6: Distribution of Wire Capacitance Uncertainties at Cell Placement Design Phase.

adjacencies. Recently the proportion of the coupling capacitance between adjacent wires increases, which results in the increase of uncertainties at placement phase. The ratio of the estimated capacitance at placement phase compared with the capacitance of the final layout is evaluated using a 32-bit CPU circuit(about 13k cells). Fig. 4.6 shows the distribution of the estimation error of the wire capacitance at cell placement phase. Even when the cell place is fixed, there is the wire capacitance uncertainty of which the standard deviation is 25% of the estimated capacitance.

RC extraction tools have a certain amount of estimation errors. The amount of errors in capacitance extraction may vary depending on the used algorithm (2D, quasi 3-D, 3D etc.) as well as on the complexity of the interconnect structures under extraction. It is not easy to estimate the uncertainty in the extraction, but the standard deviation of 10% is thought to be a reasonable guess.

Summary of Uncertainties

From the above discussion, the assumption of the delay uncertainty sources is summarized as follows.

Manufacturing Variability The delay time of each gate fluctuates such that the mean delay is its typical delay time and the standard deviation is 8% of its typical delay.

Extraction Error The extracted wire capacitance has the error of which σ is 10% of the extracted value.

Uncertainty at Placement The wire capacitance estimated at cell placement design phase has the uncertainty of wire capacitance. The mean is the estimated value and the standard deviation is 25% of the estimated value.

Results

The worst-case delay time is evaluated as x_1 in Eq. (4.9), which corresponds to mean+3σ in a normal distribution, under each uncertainty source. The result of statistical timing analysis is shown in Table 4.2. The column under "Typ. Delay" is the circuit delay time with no delay fluctuation. The columns "Manufacturing Variability", "Extraction Error" and "Uncertainty at Placement" correspond to the results under each uncertainty source respectively. The columns "Delay" are the worst-case circuit delay time with delay fluctuation. "Inc." means the percentage of the delay time increase caused by delay fluctuation. "MV+EE" is the result under both manufacturing variability and extraction error. This situation corresponds to the final delay evaluation of the completed circuit using an accurate RC extraction tool. "MV+EE+UP" means the situation that the circuit delay is estimated at cell placement design phase. So, the result under all three fluctuation sources is listed below "MV+EE+UP". The column "CPU Time" represents the CPU Time for timing analysis on Alpha Station.

Due to manufacturing variability, extraction error, and uncertainty at placement, the worst-case circuit delay increases by 9.2%, 2.4% and 8.2% on average from the delay without fluctuation, respectively. The amount of increase varies from circuit to circuit under the same uncertainty sources. For example, the increase caused by the uncertainty at placement ranges from 4.4% to 14.7%, which indicates that the impact of uncertainty is considerably different in each circuit.

In the evaluation of the circuit from the final layout("MV+EE"), the delay increases by 9.8% on average from the typical delay. This result indicates that the circuit design does not succeed without the consideration of local delay uncertainties. In the case of the delay estimation at cell placement design phase("MV+EE+UP"), there is a possibility that the delay time increases by 13.4%.

4.5.3 Delay and Power Optimization under Wire Capacitance Uncertainties

The delay and power optimization results under wire capacitance uncertainties is demonstrated. The wire capacitance is assumed to fluctuate according to a normal distribution. The mean is the value used in the logic synthesis. The standard deviation is 50% of its mean value, which corresponds to the delay uncertainties of 20% or less.

First, the delay optimization results is shown. The circuits is optimized to minimize the delay time. Please note that the initial circuits used for this experiment are synthesized and optimized for minimizing the circuit delay under the deterministic delay model. Table 4.3 shows the delay optimization results. "Initial" and "Optimized" correspond to the initial circuit before the optimization and the circuit optimized for delay minimization respectively. "Area" is calculated as the sum of the cell area. The proposed method reduces the delay time by 8.4% on average. This result shows that the circuit optimized without the consideration of fluctuations is not optimal. The optimization method considering statistical variation is effective for getting better circuits.

Next, the power optimization results(Table 4.4) are shown. The power dissipation is optimized under the delay constraints of the initial delay time. The proposed method reduces power dissipation by 9.3% on average and area by 5.1% without the increase of delay time.

Table 4.2: Delay Fluctuation.

Circuit	Typ. Delay (ns)	Manufacturing Variability		Extraction Error		Uncertainty at Placement		MV+EE		MV+EE +UP		CPU Time (s)	#Gates
		Delay (ns)	Inc. (%)	Delay (ns)	Inc. (%)	Delay (ns)	Inc. (%)	Delay (ns)	Inc. (%)	Delay (ns)	Inc. (%)		
C432_A	4.48	4.89	9.2	4.52	0.9	4.76	6.3	4.90	9.4	4.98	11.1	0.03	178
C432_B	4.97	5.39	8.5	5.02	1.0	5.19	4.4	5.40	8.7	5.49	10.5	0.03	154
C432_C	5.91	6.37	7.8	6.00	1.5	6.21	5.1	6.40	8.3	6.52	10.3	0.03	144
C432_D	6.92	7.60	9.8	7.08	2.3	7.41	7.1	7.63	10.3	7.82	13.0	0.03	130
C3540_A	6.71	7.32	9.1	6.77	0.9	7.04	4.9	7.32	9.1	7.39	10.1	0.17	871
C3540_B	7.18	7.78	8.4	7.25	1.0	7.51	4.6	7.79	8.5	7.89	9.9	0.16	835
C3540_C	7.97	8.61	8.0	8.13	2.0	8.56	7.4	8.63	8.3	8.94	12.2	0.16	703
C3540_D	8.92	9.59	7.5	9.06	1.6	9.46	6.1	9.62	7.8	9.86	10.5	0.16	657
C5315_A	6.00	6.60	10.0	6.13	2.2	6.44	7.3	6.62	10.3	6.82	13.7	0.28	1001
C5315_B	6.97	7.61	9.2	7.15	2.6	7.54	8.2	7.65	9.8	7.89	13.2	0.25	946
C5315_C	7.98	8.69	8.9	8.17	2.4	8.59	7.6	8.73	9.4	9.01	12.9	0.25	932
C5315_D	8.90	9.65	8.4	9.12	2.5	9.62	8.1	9.70	9.0	10.04	12.8	0.26	919
C7552_A	4.84	5.33	10.1	4.93	1.9	5.16	6.6	5.34	10.3	5.47	13.0	0.29	1339
C7552_B	5.02	5.49	9.4	5.11	1.8	5.34	6.4	5.51	9.8	5.63	12.2	0.29	1248
C7552_C	5.99	6.49	8.3	6.08	1.5	6.43	7.3	6.52	8.8	6.72	12.2	0.31	1127
C7552_D	6.95	7.56	8.8	7.12	2.4	7.52	8.2	7.61	9.5	7.86	13.1	0.32	1087
alu4_A	3.31	3.63	9.7	3.37	1.8	3.58	8.2	3.64	10.0	3.74	13.0	0.24	1386
alu4_B	3.99	4.40	10.3	4.11	3.0	4.38	9.8	4.43	11.0	4.61	15.5	0.26	1219
alu4_C	4.95	5.35	8.1	5.10	3.0	5.46	10.3	5.40	9.1	5.67	14.5	0.31	1184
alu4_D	5.83	6.30	8.1	6.06	3.9	6.50	11.5	6.37	9.3	6.72	15.3	0.34	1167
des_A	3.60	4.02	11.7	3.70	2.8	3.98	10.6	4.04	12.2	4.20	16.7	1.00	2252
des_B	3.98	4.44	10.6	4.16	4.5	4.51	13.3	4.47	12.3	4.75	19.3	1.26	1927
des_C	4.96	5.50	10.9	5.23	5.4	5.69	14.7	5.58	12.5	5.94	19.8	1.25	1769
des_D	5.91	6.49	9.8	6.14	3.9	6.62	12.0	6.55	10.8	6.93	17.3	0.87	1714
Average	-	-	9.2	-	2.4	-	8.2	-	9.8	-	13.4	-	-

4.6 Conclusion

In this chapter, a performance optimization method based on statistical static timing analysis is proposed. A technique that improves the accuracy of the worst-case delay analysis is developed. A new measure that represents the timing criticality at each gate is devised, and the optimization algorithm utilizing the measure is shown. The accuracy of the worst-case delay calculation is verified experimentally. The maximum estimation error is within 3%. The delay fluctuation is evaluated under some of the delay uncertainty sources. The results also demonstrate that the proposed method can reduce delay and power dissipation from the circuits optimized without the consideration of fluctuation.

Table 4.3: Delay Optimization.

Circuit	Initial			Optimized				CPU
	Delay (ns)	Area (mm^2)	Power (mW)	Delay (ns)	Delay Reduction(%)	Area (mm^2)	Power (mW)	Time (s)
C432_A	5.22	0.017	33	4.86	6.9	0.018	34	12
C3540_A	7.60	0.083	147	7.00	7.9	0.088	159	462
C5315_A	7.17	0.089	138	6.39	10.9	0.093	147	260
C7552_A	5.58	0.134	234	5.19	7.0	0.138	243	695
alu4_A	3.96	0.122	244	3.65	7.8	0.126	254	224
des_A	4.56	0.214	383	4.11	9.9	0.214	389	2836
Average	-	-	-	-	8.4	-	-	-

Table 4.4: Power Optimization.

Circuit	Initial			Optimized				CPU
	Delay (ns)	Area (mm^2)	Power (mW)	Area (mm^2)	Area Reduction(%)	Power (mW)	Power Reduction(%)	Time (s)
C432_A	5.22	0.017	33	0.016	5.9	29	12.1	3
C3540_A	7.60	0.083	147	0.079	4.8	135	8.2	100
C5315_A	7.17	0.089	138	0.087	2.2	131	5.1	79
C7552_A	5.58	0.134	234	0.126	6.0	209	10.7	409
alu4_A	3.96	0.122	244	0.116	4.9	220	9.8	290
des_A	4.56	0.214	383	0.199	7.0	346	9.7	5447
Average	-	-	-	-	5.1	-	9.3	-

Chapter 5

Post-Layout Transistor Sizing for Power Reduction in Cell-Base Design

This chapter discusses a transistor sizing method that down-sizes MOSFETs inside a cell to eliminate redundancy of cell-based circuits as much as possible. The proposed method reduces power dissipation of detail-routed circuits while preserving interconnects. The effectiveness of the proposed method is experimentally evaluated using 5 circuits. The power dissipation is reduced by 77% maximum and 65% on average without delay increase.

5.1 Introduction

Cell-base design has a well-established framework for the development of ASICs, and has been widely adopted. On the other hand, cell-based circuits inherently contain redundancy, for example, in power dissipation. In this chapter, a post-layout transistor sizing method for power reduction is proposed. The proposed method aims to reduce the redundancy of cell-base design and to obtain high performance circuits close to full-custom quality while keeping the cell-base design framework. MOSFETs inside a cell is down-sized continuously, and the corresponding cell layout is generated on the fly. The cell layout generation system used in the proposed method does not change the location of input and output pins while the transistor widths inside a cell are varied[27]. Exploiting this feature, the proposed method can optimize detail-routed circuits, without any modifications of interconnects, using the precise wire capacitance values extracted from the detail-routed circuits.

Many transistor sizing methods for delay and power optimization have been proposed[1, 3, 64, 2, 5]. These methods need to derive the delay time of each cell at any MOSFET size. Refs.[1, 3, 64] utilize Elmore delay model. In this delay model, the optimal solution of the problem can be obtained using a simple variable-transformation method. However, the accuracy of the delay model is not high enough, and hence the optimized circuits may violate the delay constraints. In Refs. [2, 5], the cell delay is approximated as a linear function of the cell size, and transistor sizing is formulated as a linear optimization problem. This method also can obtain the optimal solution of the formulated problem. However, the linearization

of the cell delay may introduce errors in timing analysis.

Recently, the delay time due to wire capacitance occupies a considerable part of the total circuit delay. Many of the previous transistor sizing methods[1, 3, 2, 5] concentrate on circuit-level optimization, and the consideration on layout is not enough. When the optimization result is applied to the layout, routing is affected, i.e. wire capacitances in the resulting layout become different from the initial circuit before transistor sizing. The variation of wire capacitance may cause a violation of delay constraints. In Ref. [64], transistor sizing, re-routing and compaction techniques are performed to the circuit repeatedly for better consideration on layout. In a DSM process, coupling capacitances between adjacent interconnects in the same metal layer or two successive metal layers become dominant. The accurate capacitance evaluation of all the interconnects influenced by re-routing and compaction becomes computationally intensive and hence the repeated evaluation inside the optimization loop may become impractical.

The proposed method handles detail-routed circuits designed in cell-base design style. The proposed method down-sizes MOSFETs inside a cell for power reduction without any modifications of wiring using accurate values of wire capacitance. The proposed method uses a cell layout generation system called VARDS[27] that can generate cell layout with variable transistor width while keeping the location of terminals unchanged. In order to get the accurate cell delay time, the proposed method utilizes four-dimensional look-up tables with four variables; gate widths of PMOS and NMOS transistors, input transition time, and load capacitance.

This chapter is organized as follows. Section 5.2 explains the post-layout transistor sizing method. Cell layout generation, cell delay model, and transistor sizing algorithms are discussed. Section 5.3 demonstrates some experimental results. Finally, Section 5.4 concludes the discussion.

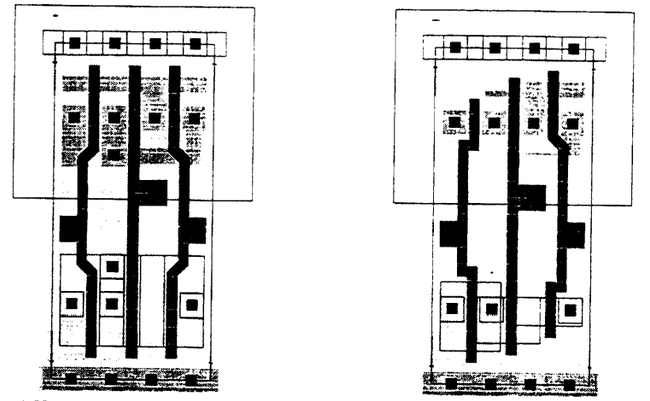
5.2 Post-Layout Transistor Sizing

This section explains a transistor sizing method for power reduction preserving interconnects. First cell layout generation for post-layout transistor sizing is discussed. Next, a cell delay model that can calculate delay time for any PMOS and NMOS transistor sizes is shown. Then, the noise margin constraints that guarantee the correct behavior of the circuits are discussed. Finally, a transistor sizing algorithm for power reduction is explained.

5.2.1 Cell Layout Generation

In order to apply the optimization result to the layout without any modifications of interconnects, the following features are required for cell layout generation.

- Each transistor width can be varied easily and flexibly.
- The location of each pin is fixed even when transistor widths are varied.



(a) All transistor widths are the maximum.

(b) Every transistor width is different.

Figure 5.1: Examples of AOI21 Cell Layout.

The fixed locations of input/output pins are needed to preserve interconnects. A cell layout generation system VARDS, which satisfies the above two requirements, has been proposed[27]. Fig. 5.1 shows an example of AOI21 cells whose height is 9 interconnect pitches. The AOI21 cell in Fig. 5.1(a) is generated such that all transistor widths are the maximum. Fig. 5.1(b) is an example that every transistor width is different.

5.2.2 Cell Delay Model

In the proposed method, PMOS and NMOS transistors inside a cell are resized separately. The proposed method hence requires a cell delay model that has four variables, W_p, W_n, tt, and cl, where $W_p(W_n)$ is the gate width of PMOS(NMOS) transistor, tt is the transition time of the input signal, and cl is the capacitive load. Four-dimensional look-up tables with four variables W_p, W_n, tt, and cl are built beforehand using a circuit simulator. Cell delay time is derived from the look-up tables using the following three-step interpolation(Fig. 5.2). In the case of a multi-stage cell, the cell is divided into single-stage cells, and the delay time of each single-stage cell is calculated.

Step1: Find four neighboring points(P_1, P_2, P_3, P_4) around the evaluation point(P_{ev}), in two-dimensional W_p-W_n space.

Step2: Calculate the delay time at each point of P_1, P_2, P_3, P_4 using Eq. (5.1) in two-dimensional tt-cl space .

Step3: Interpolate rise/fall delay time using Eq. (5.2/5.3) in W_p-W_n space from the four values at P_1, P_2, P_3, P_4 calculated at **Step2**.

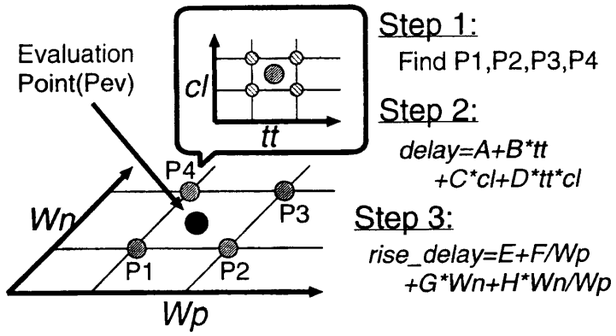


Figure 5.2: Derivation of Cell Delay.

$$delay = A + B \cdot tt + C \cdot cl + D \cdot tt \cdot cl, \tag{5.1}$$

$$rise\_delay = E + F \cdot \frac{1}{W_p} + G \cdot W_n + H\frac{1}{W_p} \cdot W_n, \tag{5.2}$$

$$fall\_delay = I + J \cdot W_p + K \cdot \frac{1}{W_n} + L \cdot W_p\frac{1}{W_n}, \tag{5.3}$$

$$energy = M + N \cdot W_p + O \cdot W_n + P \cdot W_p \cdot W_n, \tag{5.4}$$

where, $A, B, ..., P$ are coefficients to be determined such that the four values of the neighboring points are assigned to each interpolation equation. The transition time of the output signal is calculated similarly. In the case of the dissipated energy, Eq. (5.4) is used for the interpolation at **Step3**.

5.2.3 Noise Margin Constraints

Adequate amounts of noise margins are important to ensure the correct behavior of the circuits. The noise margins are defined as $NM_H = V_{OH} - V_{IH}$ and $NM_L = V_{IL} - V_{OL}$. The noise margin depends on the ratio β_R, which is expressed as $\beta_n/\beta p$, where $\beta_{n(p)}$ is the n(p)-device transconductance. The range of β_R that guarantees proper noise margins is calculated. The upper bound $\beta_{R(max)}$ can be derived from the following two equations[65, 66].

$$V_{IL} = \frac{2V_{out} - V_{DD} + V_{Tp} + \beta_{R(max)}V_{Tn}}{1 + \beta_{R(max)}}, \tag{5.5}$$

$$\begin{aligned}\beta_{R(max)}(V_{IL} - V_{Tn})^2 &= -(V_{out} - V_{DD})^2 \\ &+2(V_{IL} - V_{DD} - V_{Tp})(V_{out} - V_{DD}).\end{aligned} \tag{5.6}$$

Similarly, the lower bound $\beta_{R(min)}$ can be obtained from the following two equations.

$$V_{IH} = \frac{\beta_{R(min)}(2V_{out} + V_{Tn}) + V_{DD} + V_{Tp}}{1 + \beta_{R(min)}}, \tag{5.7}$$

$$\begin{aligned}\beta_{R(min)}[2(V_{IH} - V_{Tn})V_{out} - V_{out}^2] \\ = (V_{IH} - V_{DD} - V_{Tp})^2,\end{aligned} \tag{5.8}$$

where V_{Tp}, V_{Tn} are the threshold voltages of PMOS and NMOS transistors. The proposed method resizes PMOS and NMOS transistors for power reduction within the range of $\beta_{R(min)} < \beta_R < \beta_{R(max)}$.

5.2.4 Transistor Sizing Algorithm

A transistor sizing algorithm for power reduction based on sensitivity calculation is devised. The proposed algorithm executes iterative optimization that decreases δ_{size} gradually, where δ_{size} is a variable that represents the amount of transistor width reduced in a single iteration.

Step1: Set δ_{size} to an initial value.

Step2: If δ_{size} is smaller than a pre-defined value, the optimization procedure finishes.

Step3: At each cell, evaluate the sensitivity, i.e. the amount of power reduction when the transistor widths decrease by δ_{size}. If the violations of noise margin or transition time constraints occur, sensitivity calculation is not performed.

Step4: Select the cell with the best sensitivity. If there are no cells with positive sensitivity, halve δ_{size} and go back to **Step2**.

Step5: Decrease the transistor widths of the selected cell by δ_{size}, and update the timing information of the cells affected by the down-sizing. If delay violation occurs, cancel the down-sizing.

Step6: Find the cell with the next best sensitivity. If there are no cells with positive sensitivity, go back to **Step3**. Otherwise, go back to **Step5**.

First, the above algorithm is executed for power reduction such that PMOS and NMOS transistors are resized simultaneously with the same β_n/β_p ratio. Next power dissipation is optimized resizing PMOS and NMOS transistors independently, and then the final optimization result is obtained.

5.3 Experimental Results

In this section, some experimental results are shown. First the accuracy of the cell delay model based on look-up tables is demonstrated. Next the power optimization results are shown.

Cell layouts are generated using VARDS[27] in a 0.35μm process with three metal layers. The cell height is 13 interconnect-pitches, and the size ratio of PMOS and NMOS transistors is 1. In transistor sizing, MOSFETs are down-sized within the range that VARDS can generate cell layouts. The maximum transistor width of standard driving-strength(x1) cells

Table 5.1: Average Error of Cell Delay Model Based on Look-up Tables.

Cell	Transition	Variables of Interpolation		
		W_p, W_n, tt, cl	tt, cl (W_p, W_n fixed)	W_p, W_n (tt, cl fixed)
INV	rise	0.003ns 1.9%	0.002ns 1.4%	0.001ns 1.0%
	fall	0.004ns 1.3%	0.002ns 0.9%	0.002ns 0.4%
NAND2	rise	0.003ns 2.1%	0.002ns 1.5%	0.001ns 0.9%
	fall	0.005ns 1.0%	0.002ns 0.6%	0.003ns 0.4%
NOR2	rise	0.002ns 1.2%	0.001ns 0.8%	0.001ns 0.6%
	fall	0.005ns 1.2%	0.002ns 0.7%	0.003ns 0.5%

is 6.2μm, and the value of W/L is 15.5. The transistor width can be reduced to 0.9μm. Reference [26] reports that the optimal value of W/L is around 20. The transistor width of the library used in the experiments is smaller than the reported value.

5.3.1 Accuracy of Cell Delay Model

First the accuracy of the cell delay model is examined. INV, 2-input NAND and 2-input NOR cells of standard driving-strength(x1) are used for this experiment. In the case of NAND and NOR cells, the characteristics of the input pin that is close to the output terminal are evaluated. The delay time derived by the interpolation in Sec. 5.2.2 is compared with the delay time evaluated by circuit simulation at the following 6561 points. The gate widths of PMOS and NMOS transistors (W_p, W_n) are varied to 0.9, 1.2, 1.5, 2.0, 2.5, 3.2, 4.0, 5.0, and 6.2μm, respectively. The evaluation points of the input transition time (tt) are 0.02, 0.125, 0.25, 0.375, 0.5, 0.65, 0.8, 1.0, and 1.2ns, also the points of load capacitance (cl) are 0.005, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2, 0.35 and 0.5pF. The combinations of W_p and W_n that the noise margin becomes smaller than $0.25V_{DD}$ are excluded. When the absolute value of the delay time is extremely small, the relative error becomes meaninglessly large while absolute error is sufficiently small. The relative error is not hence calculated when the delay time is less than 0.01ns. The size of look-up tables is 5x5x5x5. Table 5.1 shows the error of the cell delay model. The interpolation error of the delay time derived in W_p-W_n space is comparable with the error calculated in tt-cl space. It therefore can be seen that the interpolation in W_p-W_n space by Eqs. (5.1) and (5.2) is reasonable. The average error of the delay time calculated from 4-dimensional look-up tables of W_p, W_n, tt, and cl is less than 2%. Compared with the interpolation in tt-cl space, the average error increases by 0.5%.

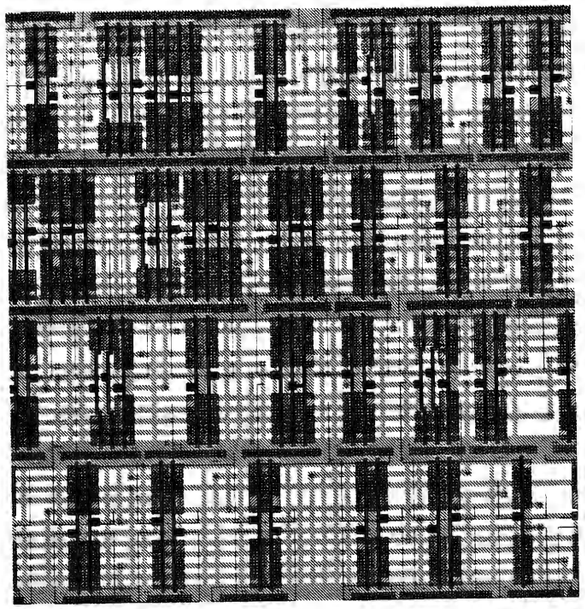
5.3.2 Power Optimization Results

The power optimization results are shown. The circuits used for the experiments are an ALU in a DSP for mobile phone[67] (dsp_alu) and the circuits included ISCAS85 and LGSynth93 benchmark sets (C3540, alu4, C7552, des). These circuits are synthesized under two different constraints [56]: minimizing the circuit delay, and minimizing the circuit area. Also two transition time constraints, 0.5ns and 1.0ns are given. Thus, each circuit is synthesized under four different constraints in total. The layouts of the synthesized circuits are generated, and the wire capacitance values extracted from the layouts for transistor sizing are utilized. The circuit scale is 943 to 12460 cells. The cell library used for generating initial circuits includes six varieties in driving-strength for INV and BUF (x1, x2, x3, x4, x6 and x8). In the case of NAND2, NAND3, AND2, AND3, NOR2, NOR3, OR2, OR3, AOI21, OAI21 cells, there are four varieties(x1, x2, x3, x4). The circuit delay time is evaluated by a transistor-level static timing analysis tool[68], and the power dissipation is estimated by a transistor-level power simulator[41]. The input patterns are randomly generated with a transition probability of 0.5. The number of applied patterns is 100, which is the adequate number for power estimation at circuit level[43]. The cycle time of the input patterns is 100ns.

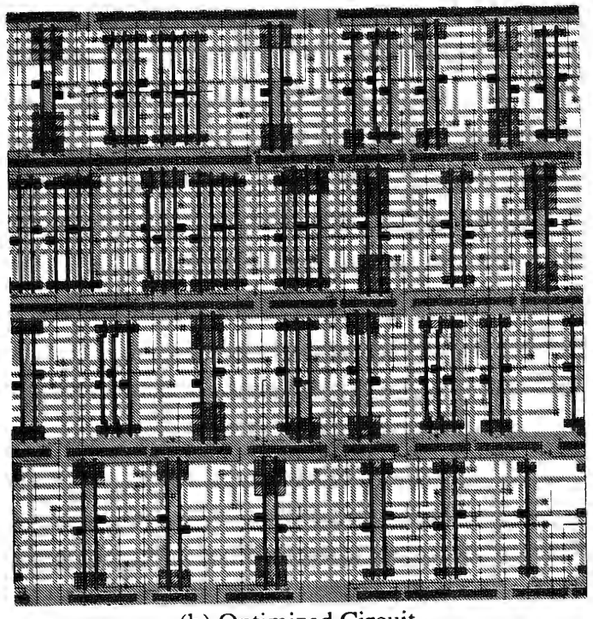
Power dissipation is optimized under the delay constraints of the initial circuits' delay time. The initial value of δ_{size} in the optimization algorithm(Sec. 5.2.4) is 12.4μm, and the termination value is 0.1μm. The constraints that the noise margin is larger than $0.25V_{DD}$ are given. Table. 5.2 shows the power optimization results. The column "Total Width" represents the sum of the gate widths of MOSFETs in the circuit. "CPU Time" represents the CPU time required for power optimization on an Alpha Station. The proposed method reduces power dissipation by 77% maximum and 65% on average. The total transistor width is reduced to 25% of the initial circuits. The power reduction in small circuits is larger than the one in large circuits, because large circuits usually have heavier wire load. In the case of the largest circuit dsp_alu, the power dissipation is reduced by about 50%. In some circuits, the circuit delay increases though the initial delay time is given as the delay constraints. One reason is that the optimized circuits become sensitive to the error of cell delay model, which will be discussed in Chapter 6. Further examination of the reasons is required, considering the accuracy of the delay calculation tool as well.

The following discussion examines the optimization result of des circuit generated for minimizing circuit delay under the transition time constraint of 0.5ns. Fig. 5.3(a) shows a part of the initial layout. Fig. 5.3(b) corresponds to the transistor-sized layout of the same location. The transistor sizes inside cells become different in instance by instance. PMOS and NMOS transistors inside each cell are resized separately. Also the routing is perfectly preserved. The proposed method generates cell layouts on the fly according to the optimization results, and replaces cells without any interconnect modifications.

First the relationship between the amount of power reduction and the increase of driving-strength varieties is demonstrated. Halving δ_{size} in the optimization algorithm(Sec. 5.2.4) corresponds to halving the intervals of driving-strength and increasing driving-strength varieties twofold. The driving-strength varieties are classified into 10 levels(Table 5.3). Fig. 5.4



(a) Initial Circuit.



(b) Optimized Circuit.

Figure 5.3: A Part of Layout(des, Fastest, Transition Time Constraint 0.5ns).

Table 5.2: Power Optimization Results(Cell Height: 13 Interconnect Pitches).

Circuit	Transition Time Constraints (ns)	Design Constraints	Initial Circuits			Optimized Circuits				CPU Time (s)	#cells
			Delay (ns)	Power (mW)	Total Width (mm)	Delay (ns)	Power (mW)	Power Reduction (%)	Total Width (mm)		
C3540	0.5	Fastest	5.3	6.1	27.0	5.3	1.6	74	5.64	100	1039
		Min-Area	6.9	4.8	21.8	7.1	1.3	73	4.33	62	943
	1.0	Fastest	4.4	6.7	26.7	4.6	1.7	75	5.75	111	1207
		Min-Area	6.1	3.5	13.0	6.5	0.9	74	2.54	37	895
alu4	0.5	Fastest	2.9	5.1	42.6	3.1	1.9	63	12.6	213	1613
		Min-Area	4.0	4.2	33.8	4.1	1.4	67	8.70	145	1403
	1.0	Fastest	2.2	4.6	33.2	2.5	1.9	59	10.4	200	1568
		Min-Area	3.6	3.1	18.7	3.7	1.1	65	4.53	76	1361
C7552	0.5	Fastest	4.2	14.5	49.0	4.4	3.4	77	9.74	279	1995
		Min-Area	6.2	12.7	37.0	6.5	3.0	76	7.00	160	1687
	1.0	Fastest	3.3	14.1	44.6	3.5	3.2	77	9.21	275	2043
		Min-Area	5.1	8.5	22.1	5.1	2.1	75	4.38	97	1619
des	0.5	Fastest	3.2	14.4	84.7	3.4	5.1	65	20.6	925	3414
		Min-Area	4.2	11.1	63.4	4.5	4.3	61	15.3	560	2908
	1.0	Fastest	2.7	13.4	68.0	2.8	5.3	60	18.8	772	3327
		Min-Area	3.4	8.5	41.1	3.7	3.7	56	10.6	371	2859
dsp_alu	0.5	Fastest	8.8	79.8	347	9.4	37.1	54	115	20304	12547
		Min-Area	18.1	75.9	299	17.7	39.9	47	109	15436	11765
	1.0	Fastest	7.2	66.2	235	8.1	28.2	57	65.7	9203	12460
		Min-Area	15.3	54.3	169	15.8	26.3	52	44.9	4831	10892
Average	-	-	-	-	-	-	-	65	-	-	-

Table 5.3: Driving-Strength Level.

Level	#driving-strength varieties(INV)	δ_{size} (μm)	PN ratio
Level 0	6 (Initial)	-	-
Level 1	11	12.4	Fixed
Level 2	23	6.2	Fixed
Level 3	44	3.1	Fixed
Level 4	85	1.55	Fixed
Level 5	166	0.775	Fixed
Level 6	332	0.388	Fixed
Level 7	659	0.194	Fixed
Level 8	1314	0.097	Fixed
Level 9	1.7M	0.097	Varied

indicates the relationship between power dissipation and driving-strength level. The power dissipation is reduced as the driving-strength varieties increase.

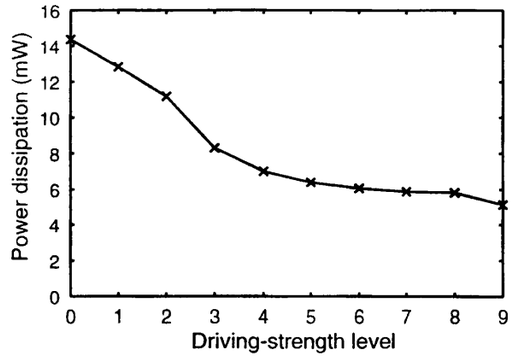


Figure 5.4: Relationship between Power Dissipation and Driving-Strength Varieties(des, Fastest, Transition Time Constraint 0.5ns).

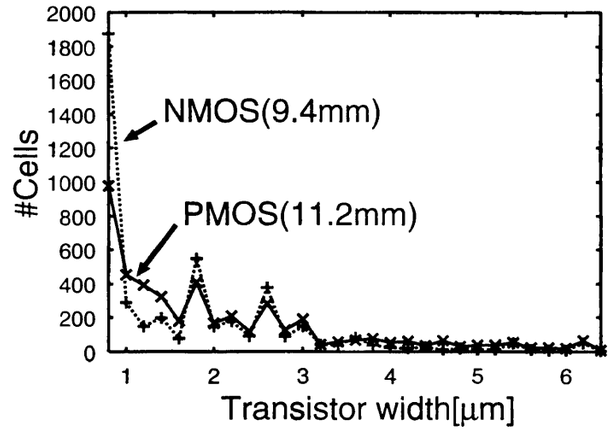


Figure 5.5: Distribution of Transistor Widths(des, Fastest, Transition Time Constraint 0.5ns).

Next the distributions of transistor widths in the optimized circuit are shown(Fig. 5.5). The transistor width of a standard driving-strength(x1) cell is 6.2μm, and the transistor width can be reduced to 0.9μm. Many MOSFETs are down-sized close to the lower limit of 0.9μm. Compared with PMOS transistors, the gate widths of NMOS transistors are small. The sum of PMOS gate widths is 11.2mm, which is 19% larger than the sum of NMOS gate widths(9.4mm).

Fig. 5.6 expresses the slack distributions of the initial and optimized circuits. By transistor sizing, the number of the cells with 0 or almost 0 slack increases drastically. The sum of slack in the optimized circuit is 1241ns, whereas the sum of slack in the initial circuit is 3122ns. The total slack is reduced by 60%.

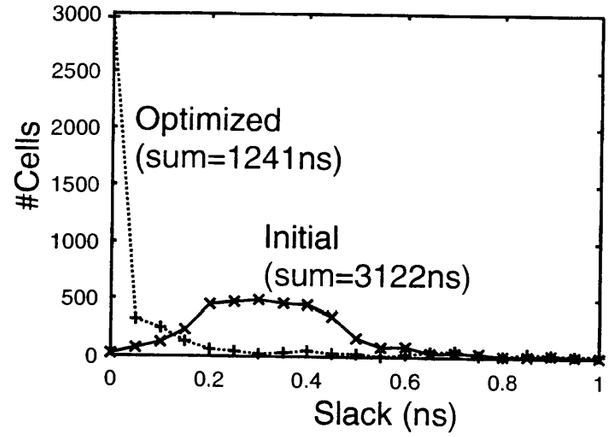


Figure 5.6: Distribution of Slack(des, Fastest, Transition Time Constraint 0.5ns).

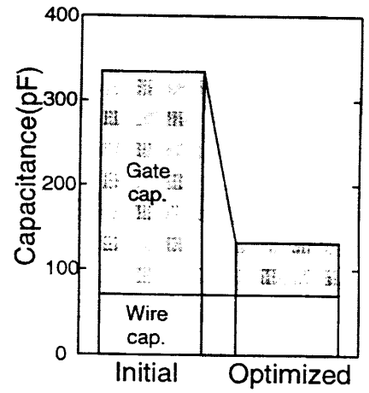


Figure 5.7: Capacitance Reduction(des, Fastest, Transition Time Constraint 0.5ns).

Then the capacitance reduction in the circuit is demonstrated(Fig. 5.7). The proposed method does not modify any interconnects, so wire capacitance does not change. The gate capacitance of MOSFETs is reduced by 77%, which results in 61% reduction of the total capacitance.

The peak current reduction is shown. 100 input patterns are given, and the peak current is evaluated at each time-step within a cycle. Fig. 5.8 indicates the peak current of the initial and optimized circuits. The horizontal axis represents the time within a cycle of 3.4ns. The peak current is reduced by 74%. Path-balancing effect of the proposed method contributes to the peak current reduction, as well as gate capacitance reduction. The transition timing of each cell is well distributed throughout a cycle. Reducing the peak current is effective to

avoid IR drop problem. Also, the current reduction is a useful way to evade electromigration. The mean time to failure(MTF) of electromigration t_f is expressed as follows[69].

$$t_f = AW^p L^q J^{-n} exp(E_a/kT), \tag{5.9}$$

where J is current density, E_a is activation energy, W is the width of metal, L is the length, and n is a constant close to 2. The current reduction of 74% increases MTF 15 times. Thus, the proposed method can increase the tolerance to IR drop and electromigration problems, and contribute to high-reliability LSI design.

The power optimization results, when the initial circuits are generated using a low-power cell library, are shown. The delay time of each initial circuit is given as the delay constraint. The cell-height of this low-power library is 9 interconnect pitches, and the standard transistor size is 3.4μm. The results are shown in Table 5.4. Even when the low-power cell library is used for initial circuits, the proposed method reduces power dissipation by more than 50% on average.

5.3.3 Effectiveness of Interconnect Preservation

The proposed method optimizes a detail-routed circuit without any wiring modifications. The effectiveness of the interconnect preservation is verified. In a conventional transistor sizing methods, the layout is modified using an ECO(Engineering Change Order) technique in order to preserve the placement and wiring as much as possible. But a certain amount of variation in wire capacitance is not avoidable.

The effect of this capacitance variation is examined statistically. It is assumed that the wire capacitance varies according to a normal distribution N(m, σ) because of interconnect modifications, i.e. ECO. The mean m is the initial value used in transistor sizing, and the standard deviation σ is 20% of the initial value. The delay distribution is obtained using a Monte Carlo technique. The number of delay evaluation is 10,000. Fig. 5.9 shows the delay variation in the optimized des circuit. As you see, the interconnect modifications increase the circuit delay. The circuit whose delay time is the same with the initial circuit(3.36ns) can be hardly obtained. The circuit delay of "mean+3σ" is 3.60ns, which is larger than the delay without wiring modifications by 7%. The proposed method can avoid this delay increase, thanks to the interconnect preservation.

5.4 Conclusion

This chapter proposes a power reduction method that down-sizes MOSFETs in a cell without any interconnect modifications. The effectiveness of the proposed method is experimentally verified using 5 benchmark circuits. The power dissipation is reduced by 77% maximum and 65% on average without delay increase. It is verified that the proposed method also contributes to high-reliability LSI design.

Table 5.4: Power Optimization Results (Cell Height: 9 Interconnect Pitches).

Circuit	Transition Time Constraint (ns)	Design Constraint	Initial	Optimized	
			Power (mW)	Power (mW)	Power Reduction (%)
C3540	0.5	Fastest	5.0	1.7	66
		Min-Area	2.8	1.2	57
	1.0	Fastest	4.6	1.7	63
		Min-Area	2.1	0.84	60
alu4	0.5	Fastest	3.8	1.9	50
		Min-Area	2.7	1.4	48
	1.0	Fastest	3.5	1.8	49
		Min-Area	2.0	0.98	51
C7552	0.5	Fastest	11.0	3.9	65
		Min-Area	7.4	2.9	61
	1.0	Fastest	9.8	3.3	66
		Min-Area	5.5	2.2	60
des	0.5	Fastest	10.0	4.8	52
		Min-Area	6.9	4.0	42
	1.0	Fastest	10.9	5.2	52
		Min-Area	5.2	3.1	40
dsp_alu	0.5	Fastest	55.5	31.7	43
		Min-Area	52.2	34.0	35
	1.0	Fastest	45.5	23.7	48
		Min-Area	36.9	24.4	34
Average	-	-	-	-	52

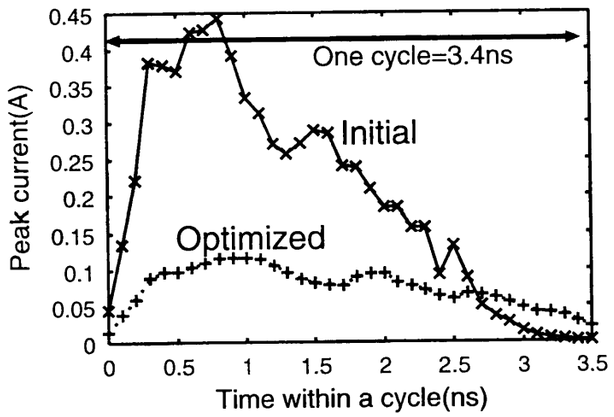


Figure 5.8: Peak Current Reduction(des, Fastest, Transition Time Constraint 0.5ns).

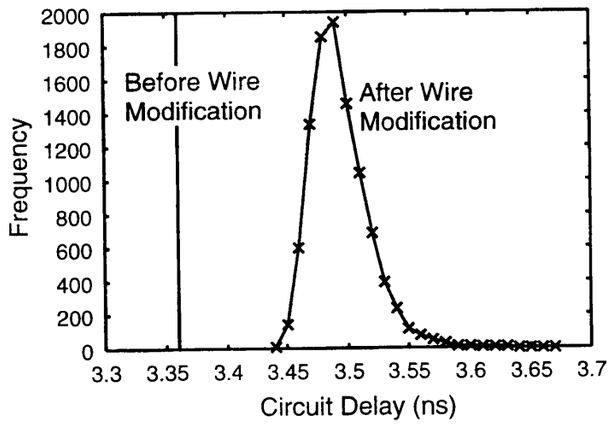


Figure 5.9: Delay Variation Caused by Interconnect Modifications(des, Fastest, Transition Time Constraint 0.5ns).

Chapter 6

Increase in Delay Uncertainty by Performance Optimization

This chapter discusses a statistical effect of performance optimization to uncertainty in circuit delay. Performance optimization has an effect of balancing the delay of each path in a circuit, i.e. the delay of long paths are shortened and the delay of short paths are lengthened. In these path-balanced circuits, the uncertainty in circuit delay, which are caused by delay calculation error, manufacturing variability, fluctuation of operating condition, etc., becomes worse by a statistical characteristic of delay. Thus, a highly-optimized circuit may not satisfy delay constraints. This chapter demonstrates some examples that uncertainty in circuit delay is increased by path-balancing, and raises a problem that performance optimization increases statistically-distributed circuit delay.

6.1 Introduction

In VLSI design, many techniques for reducing circuit delay are utilized at each design phase in order to satisfy given timing constraints. For example, division into pipeline stages, clock scheduling, logic composition, technology mapping, gate/transistor sizing, buffer insertion, wire sizing and timing driven layout synthesis are used. These methods detect the longest path and optimize the circuit for reducing the longest path delay. Recently, reducing power dissipation becomes one of the most principal subject in VLSI design. Many performance techniques, including the methods mentioned above, are hence utilized not only for delay reduction but also for reducing power dissipation. In some of these methods, blocks/cells, where timing constraints are not tight, are slowed down to reduce power consumption. Therefore, performance optimization can be regarded as a operation that shortens long paths and lengthens short paths in a circuit. The delay times of many paths in a circuit are equalized by performance optimization. This equalization is called path-balance.

There are several sources that cause uncertainties in circuit delay time, such as error in delay calculation, manufacturing variability, and fluctuation of operating conditions. The error in delay calculation includes error of delay model, diversity in signal waveforms, extraction

error of wire capacitance, and so on. The manufacturing variability consists of fluctuations in transistor characteristics and wire shapes. Also the operating condition, i.e. supply voltage and temperature, varies. Due to these sources of delay uncertainty, the delay time of each gate and wire is not a deterministic value. It necessarily has a certain probability distribution.

In the circuits optimized for performance enhancement, the delay uncertainty of each gate influences the circuit delay strongly. It is because a path-balancing operation increases the number of long paths that have possibilities to become the longest path. Due to the statistical characteristic of delay, the average value of statistically-distributed circuit delay becomes large when the number of long paths increases. This statistical effect is discussed in detail in Sec. 6.2 using a simple example. So far, this increase of statistically-distributed circuit delay caused by path-balancing has not been well discussed. Unless the statistical delay increase is considered properly, optimized circuits may not work well. In order to guarantee the circuit speed, the statistical effect of path-balancing operation needs to be understood and handled well.

In this chapter, the effect of path-balancing to uncertainty in circuit delay is examined. The influence on circuit delay is experimentally evaluated under some sources of delay uncertainty. This chapter raises a notice that performance optimization increases statistically-distributed circuit delay, and hence give a caution that more attention should be paid to the statistical effect of path-balancing in order to guarantee circuit delay time, when circuits are optimized for performance improvement. Finally a statistical static timing analysis method that is discussed in Chapter 4 is evaluated as one of solutions of this problem.

This chapter is organized as follows. Section 6.2 explains the statistical characteristic of circuit delay time. Section 6.3 shows the reason why performance optimization increases statistically-distributed circuit delay. Section 6.4 demonstrates some experimental results of statistical delay analysis and discusses the statistical effect of path-balancing to circuit delay uncertainty. Finally, Section 6.5 concludes the discussion.

6.2 Statistical Characteristic of Circuit Delay Time

The circuit delay, which is the maximum path delay time in a circuit, $D_{circuit}$ is represented as follows.

$$D_{circuit} = \max_i D_i \ \ (i = 1, 2, ..., n), \tag{6.1}$$

where D_i is the path delay time of the i-th path, and n is the number of the paths in the circuit.

Let us show a simple example of the statistical effect caused by the max operation.

$$y = \max_i x_i \ \ (i = 1, 2, ..., n). \tag{6.2}$$

Suppose that x_i is distributed according to a normal distribution N(6, 1). The distribution of y is evaluated under several values of n. Fig. 6.1 shows the distribution of y. When n increases, the average of y becomes large and the standard deviation of y becomes small. The increase of n corresponds to the increase of the number of long paths whose path delay

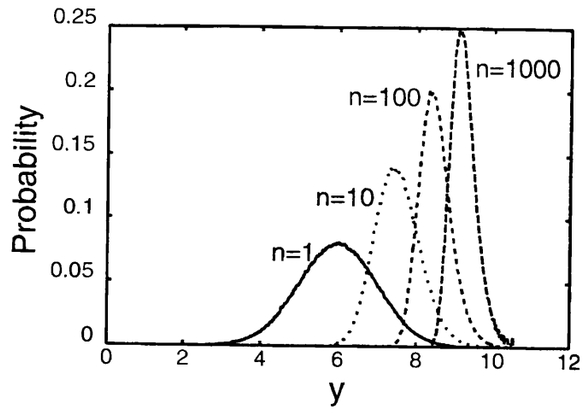


Figure 6.1: Effect of max Operation(n is varied).

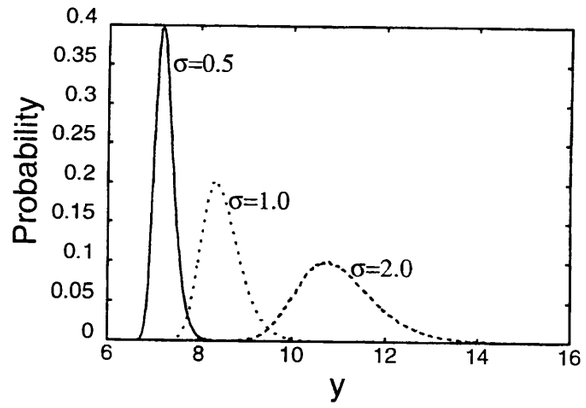


Figure 6.2: Effect of max Operation(σ is varied).

times are close to the maximum path delay. From this example, it can be seen that the distribution of $D_{circuit}$ shifts to the right, i.e. in the direction that the circuit delay increases, when the number of the long paths increases.

Another example is shown. The value n is fixed to 100, and the standard deviation σ of x_i is varied. Fig. 6.2 shows the distribution of y. When the standard deviation of x_i increases, the average and the standard deviation of y becomes large. It can be seen that the average and the standard deviation of $D_{circuit}$ become large, when the standard deviation of x_i increases.

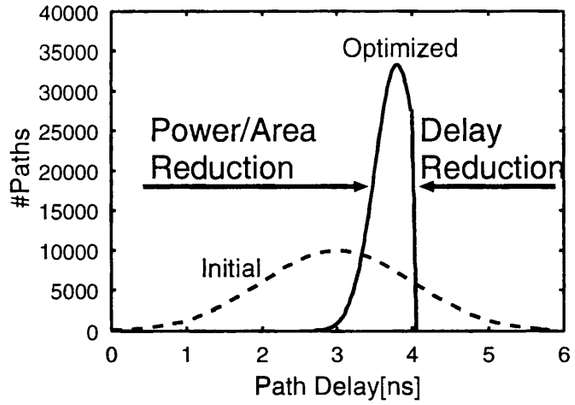


Figure 6.3: Path-Balancing Effect Caused by Performance Optimization.

6.3 Increase in Circuit Delay Uncertainty by Performance Optimization

Performance optimization generally consists of delay and power/area optimization. The delay optimization methods find long paths and optimize the circuit for reducing the longest path delay. Conversely, some of power/area optimization methods slow down the blocks/cells, where the given timing constraints are not tight, in order to reduce power dissipation, such as gate/transistor sizing[4, 48, 47, 61, 71], multiple supply voltage technique[30], multiple threshold voltage technique[72] and so on. Therefore, circuits are modified by performance optimization such that long paths are shortened and short paths are lengthened. This operation that the delay times of many paths in the circuit are equalized is called a path-balancing operation. Fig. 6.3 explains the concept of path-balancing.

The path-balancing operation increases the number of the paths whose path delays are close to the maximum path delay(Fig. 6.3). These long paths have the possibilities of becoming the longest path in the circuit. So, the increase of the number of long paths corresponds to the increase of n in Fig. 6.1. Performance optimization therefore increases statistically-distributed delay by the statistical phenomenon shown in Fig. 6.1.

6.4 Experimental Analysis

In this section, some experimental results of statistical delay analysis are shown. The results demonstrate that statistically-distributed circuit delay increases by path-balancing operation.

The ALU part of a vector processor(dsp_alu)[67] and the circuit(des) included in LGSynth93 benchmark set are used for the experiments. These circuits are synthesized and mapped by a commercial logic synthesis tool[56] under tight delay constraints. The target

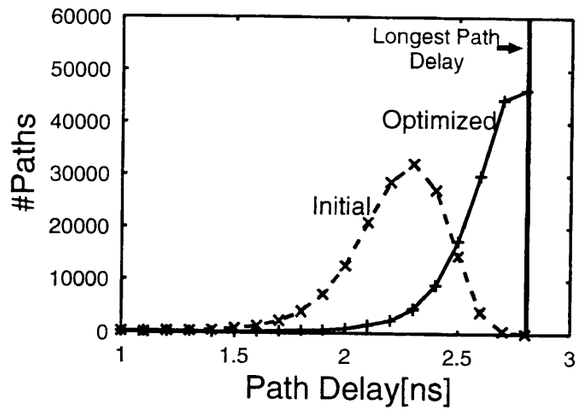


Figure 6.4: Distributions of Path Delay(des).

library is a standard cell library generated by VARDS[27] in a 0.35μm process with three metal layers. These circuits are placed and routed, and the wire capacitances are extracted from the layouts. These circuits are initial(not path-balanced) circuits. The number of gates used in dsp_alu and des are 14370 and 3837, respectively.

In order to obtain the path-balanced circuits, a transistor sizing method is utilized for performance optimization. The initial circuits are optimized by continuous transistor sizing for minimizing power dissipation under the delay constraint such that the delay does not increase from the initial value. The optimization method used for the experiments is a heuristic method that reduces power dissipation greedily based on the result of sensitivity analysis, which is discussed in Chapter 5. Figs. 6.4 and 6.5 represent the distributions of path delay in the initial and optimized circuits. It can be seen that the number of paths whose path delays are close to the longest path delay increases drastically, which corresponds to the increase of n in Fig. 6.1.

6.4.1 Analysis of Delay Uncertainty

First the impact of delay calculation error to the circuit delay uncertainty is evaluated in the initial and optimized circuits. An error model of gate delay is assumed such that the error of each gate is distributed according to a normal distribution with 3σ=10% of its typical(no error) delay. The calculation method of typical gate delay is explained in Section 5.2.2. The distribution of circuit delay is obtained by a Monte Carlo analysis. The method of Monte Carlo analysis is same with the method explained in Section 4.5. Delay fluctuation is assigned to each gate in the circuit randomly according to the given normal distribution, and evaluate the circuit delay using a static timing analysis technique. The number of delay evaluation is 10,000. The results are shown in Figs. 6.6 and 6.7. The bar labeled "Typical" represents the delay time calculated using the typical(no error) delay time for each gate. The statistically-distributed delay of the optimized circuit increases as expected. In des

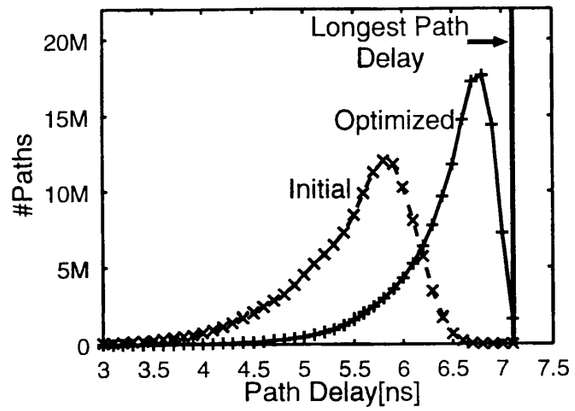


Figure 6.5: Distributions of Path Delay(dsp_alu).

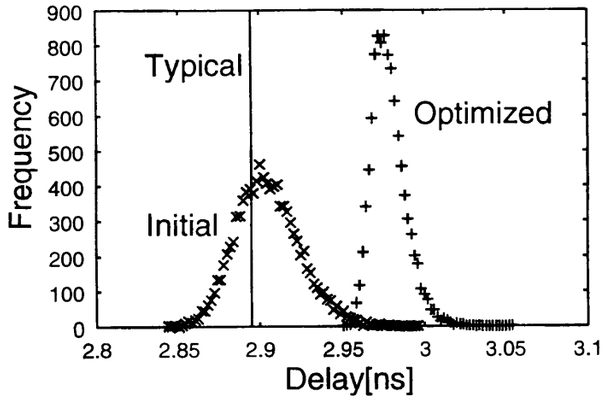


Figure 6.6: Circuit Delay Distributions under a Delay Error Model of 3σ=10%(des).

circuit(Fig. 6.6), the average delay of the optimized circuit is 2.98ns, whereas the average of the initial circuit is 2.90ns. The average delay increases by 3% by path-balancing although the circuit delay calculated from the typical delay for each gate does not change after the optimization. Also, the delay distribution of the path-balanced circuit moves far to the right of the typical delay. Therefore, in the case that the circuit is optimized considering only the typical delay, the statistically-distributed delay of the optimized circuit hardly satisfy the delay constraints.

Next, the relationships between the accuracy of gate delay and the distribution of circuit delay is examined. Three models of gate delay uncertainties are assumed such that each gate delay fluctuates normally with 3σ=5, 10 and 15% of its typical delay. In the case of a convex gate delay model for continuous transistor sizing, it is reported that 3σ of the estimation error

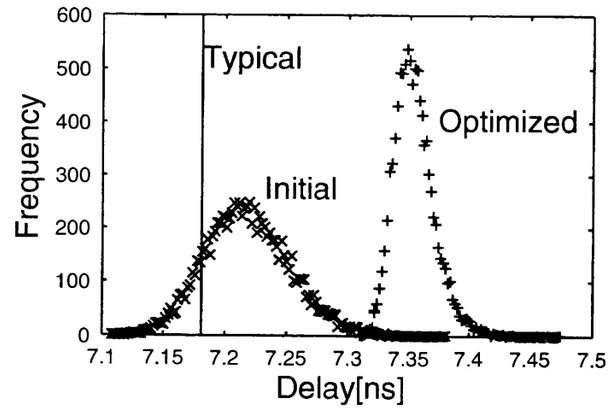


Figure 6.7: Circuit Delay Distributions under a Delay Error Model of 3σ=10%(dsp_alu).

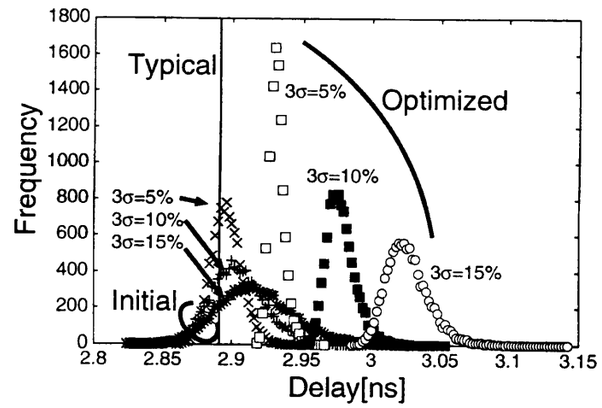


Figure 6.8: Circuit Delay Distributions under Three Delay Error Model of 3σ=5, 10, 15% (des).

in simple gates is 5 to 23% [73]. In this gate delay model, the error model of 3σ=15% might be a reasonable assumption. The model of 3σ=5% is guessed to corresponds to the delay calculation using well-designed look-up tables characterized at many points (capacitive load, input transition time, transistor sizes). Fig. 6.8 expresses the distributions of circuit delay under three error models. As the value of 3σ increases, the average and standard deviation of the circuit delay distribution becomes large, which is the same phenomenon shown in Fig. 6.2. Compared with the initial circuits, the increase of the statistically-distributed delay in the optimized circuit is large. Even when the accurate delay model with 3σ=5% is used in performance optimization, there is a distinct delay difference between the statistically-distributed delay and the typical delay in the optimized circuit.

Table 6.1: Accuracy of Statistical Static Timing Analysis in Worst-Case Delay Calculation.

Circuit	3σ of Gate Delay Error (%)	Monte Carlo	SSTA	
		Worst-Case Delay (ns)	Worst-Case Delay(ns)	Error (%)
Initial	5	2.93	2.93	0.0
	10	2.97	2.97	0.0
	15	3.01	3.02	0.3
Optimized	5	2.96	2.96	0.0
	10	3.02	3.02	0.0
	15	3.09	3.10	0.3
Average	-	-	-	0.1

Finally, the effect of circuit delay uncertainty caused by major sources of delay fluctuation is demonstrated. Three sources are considered; manufacturing variability of transistor characteristics, the extraction error of wire capacitance, and delay calculation error.

The delay calculation error is assumed to be normal which is the same model of the above experiments. The situation is supposed that each gate delay is calculated using usual look-up tables, whose number of sampling points is not large. In this case, 3σ of the cell delay error is guessed to be 10%. The magnitudes of extraction error and manufacturing variability are discussed in Section 4.5.2, and are not hence explained further. The standard deviation of extraction error is set to be 5%. As for the variability in transistor characteristics, the gate delay is assumed to fluctuate with 3σ=15%.

Fig. 6.9 shows the distributions of circuit delay under three sources of delay uncertainties. Three sources are assumed to be independent. Mean+3σ of the path-balanced circuit is 3.16ns, which is 9% larger than the typical delay. Namely, there is a possibility that the delay constraint is violated as much as 9%.

6.4.2 Worst-Case Delay Calculation

The increase of the statistically-distributed circuit delay is different between the initial and the path-balanced circuits(Figs. 6.6, 6.7, 6.8). So, setting a design margin to avoid the delay violation is difficult and seems not to be a good way. To avoid this problem, statistical delay calculation[55] and the performance optimization based on statistical delay model[60, 70] are desired. Then the statistical static timing analysis(SSTA) method[70], which is discussed in Chapter 4, is applied to the initial and optimized circuits. The circuits and the error models of gate delay are the same with those used in the previous experiment. The worst-case delay D_{worst} is evaluated, where D_{worst} is defined as x_1 in Eq. 4.9. D_{worst} corresponds to the value of $m + 3\sigma$ in a normal distribution.

Table 6.1 shows the accuracy of the statistical static timing analysis(SSTA) method discussed in Chapter 4. The column "3σ of Gate Delay Error" represents the value 3σ of gate delay uncertainties. SSTA method computes the worst-case delay D_{worst} within 0.3% error,

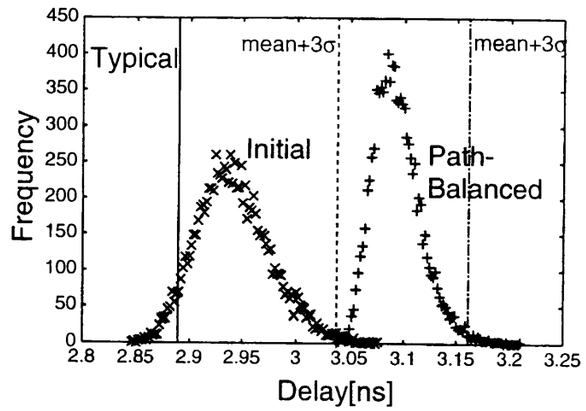


Figure 6.9: Circuit Delay Distributions under Major Delay Uncertainty Sources(des).

Table 6.2: CPU Time of Worst-Case Delay Analysis.

Monte Carlo		Statistical Static Timing Analysis
#evaluation: 10k	#evaluation: 1	
6044s	0.6s	1.9s

and the average error is 0.1%. SSTA method can calculate the worst-case delay accurately irrespective of the initial and the optimized circuits. Table 6.2 represents the comparison of CPU time needed to derive the worst-case delay. The column "Monte Carlo" corresponds to the Monte Carlo simulation whose number of delay evaluation is 10,000. Each CPU time is the average CPU time of six calculations shown in Table 6.1. SSTA method calculates the worst-case delay as more than three thousand times as fast as the Monte Carlo simulation with 10,000 delay evaluations. SSTA method requires only threefold CPU time of the Monte Carlo simulation whose evaluation number is one. In other words, SSTA needs threefold CPU time of the usual static timing analysis, although the average error of SSTA is 0.1%.

6.5 Conclusion

This chapter examines the statistical effect of path-balancing operation to uncertainty in circuit delay. Some examples that uncertainty in circuit delay is increased by path-balancing are demonstrated. This chapter raises a notice that path-balancing increases uncertainty in circuit delay, and demonstrate a problem that a highly-optimized circuit may not satisfy delay constraints.

Chapter 7

Post-Layout Transistor Sizing for Crosstalk Noise Reduction

This chapter discusses a post-layout transistor sizing method for crosstalk noise reduction. The transistors inside cells are downsized after detail-routing is completed. The proposed method estimates crosstalk noise analytically in a 2-π noise model, and optimizes crosstalk noise under delay constraints. The effectiveness of the proposed method is experimentally examined using 2 circuits. The maximum noise voltage is reduced by more than 35% without delay increase.

7.1 Introduction

Crosstalk noise problem heavily depends on interconnect structure, i.e. coupling length, spacing between adjacent wires, and coupling position, and hence many techniques of routing and interconnect optimization for crosstalk noise reduction are proposed[74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84]. Buffer insertion is also effective for noise reduction, and some methods are proposed[85, 86]. References [87, 88, 89] discuss the effectiveness of transistor sizing for crosstalk noise reduction, but practical implementations are not shown. Recently, Ref. [90] proposes a transistor sizing method for crosstalk noise reduction. In this method, crosstalk noise is estimated by Ref. [89], and circuit area is minimized under delay and crosstalk noise constraints. This transistor sizing method does not mention the layout modification after optimization. When the optimization result is applied to the layout, a certain amount of interconnects are changed, which may spoil the optimization result, or may cause a new crosstalk noise problem. In the experiments, very small and randomly-generated circuits are optimized, so the effectiveness of Ref. [90] is not clear.

Recently, several crosstalk noise models are proposed. By solving telegraph equations, the analytical formulae for peak noise is obtained[10, 91]. But these methods handle only fully-coupled interconnect structure, and can not be applied to general RC trees. In Refs. [89, 92], the aggressive wire and the victim wire are transformed into the L-type RC circuit, and the closed-form expressions of peak noise are obtained. However, the resistance of the

interconnect is not well considered in this model. In DSM technology, the wire resistance is not negligible, and the coupling location becomes one of the important factor for crosstalk noise estimation. Reference [93] assumes that the input signal is a step function, which results in overestimation of noise voltage. Recently some estimation methods that can handle distributed RC network and saturated-ramp input signal are proposed[94, 95]. In Ref. [95], moment matching technique is utilized for deriving transfer functions. Moment matching technique requires high computational cost, and hence this method is not suitable for the iterative optimization that needs to calculate crosstalk noise innumerably. Reference[88] reports that Ref. [94] overestimates crosstalk noise when the transition time of the aggressor is much larger than the victim net delay.

This chapter proposes a post-layout transistor sizing method for crosstalk noise reduction. The proposed method optimizes detail-routed circuits without any interconnect modifications. The interconnect information required for crosstalk noise estimation can be completely obtained after detail-routing. Also the optimization result of transistor sizing can be applied to the layout completely, because the proposed method utilizes the transistor sizing framework that can downsize the transistors inside cells preserving interconnects as described in Chapter 5. Thanks to these features, the proposed method reduces crosstalk noise efficiently. As for crosstalk noise estimation, a 2-π noise model with improved aggressor modeling [96] is used. The 2-π noise model is first proposed in Ref. [88]. This model can consider the location of coupling, the effect of distributed RC networks, and the slew of input signal, which are not well characterized in previous models[10, 91, 89, 92, 93, 94, 95]. However, in Ref. [88], the voltage waveform of the aggressor wire at the coupling point is approximates as a saturated ramp waveform. But in reality, the waveform is close to the exponential function, which yields estimation errors of crosstalk noise. Also the derivation of the slew of the ramp signal is not discussed. Another issue arises in the transformation of general RC trees to the 2-π noise model. Not all types of RC trees are discussed in Ref. [88]. In the proposed method, the exponential waveform is adopted as the signal of the aggressors for accuracy improvement of crosstalk noise estimation. The Elmore-like derivation method of the aggressive waveform is devised. The transformation method that can apply all types of RC trees to the 2-π noise model is developed. The optimization algorithm for crosstalk noise reduction that explores solution space effectively under delay constraints is also devised. Due to these advancements, the proposed method can estimate the crosstalk noise analytically for any RC trees, and can reduce crosstalk noise by downsizing the transistors.

This chapter is organized as follows. Section 7.2 explains the estimation method of crosstalk noise. Section 7.3 shows the optimization algorithm for crosstalk noise reduction. Section 7.4 demonstrates some experimental results. Finally, Section 7.5 concludes the discussion.

7.2 Crosstalk Noise Estimation

This section explains the estimation method of crosstalk noise. The proposed method uses the 2-π noise model[88] for crosstalk estimation. The proposed method approximates the

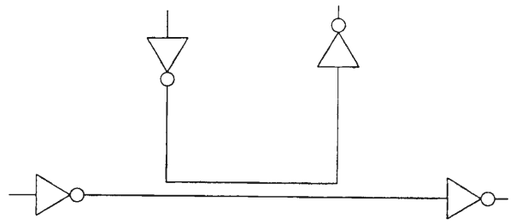


Figure 7.1: Two Coupled Interconnects.

signal of the aggressors as an exponential function for improving accuracy. The analytic waveform expressions for the aggressors and the victim are explained. The developed transformation method from practical circuits into the 2-π model is discussed.

The interconnect structure that two interconnects are partially coupled in Fig. 7.1 is considered. The partially-coupled interconnects in Fig. 7.1 are modeled as an equivalent circuit shown in Fig. 7.2. R_{v1} is the effective driver resistance of the victim net. The node n_{v2} corresponds to the middle point of the coupling interconnects. R_{v2} is the resistance between the source and n_{v2}, and R_{v3} is the resistance between n_{v2} and the sink. C_c is the coupling capacitance between the victim and the aggressor. The capacitances C_{v1}, C_{v2} and C_{v3} are represented as $C_1/2$, $(C_1+C_2)/2$, and $C_2/2+C_l$ respectively, where C_1 is the wire capacitance from the source to n_{v2}, C_2 is the wire capacitance from n_{v2} to the sink, and C_l is the capacitance of the receiver. The parameters of the aggressive wire, R_{a1}, R_{a2}, R_{a3}, C_{a1}, C_{a2}, C_{a3}, are determined similarly.

The proposed estimation method separates the victim net and the aggressive net into two equivalent circuits, as one of the approximate solutions for deriving a simple closed-form expression of noise waveform; the victim is represented as the circuit of Fig. 7.3, and the aggressor is Fig. 7.4. At the victim wire(Fig. 7.3), the aggressive wire is replaced as a voltage source. The model circuit of the victim interconnect in Fig. 7.3 becomes the same with the 2-π noise model proposed in Ref. [88],

7.2.1 Analytic Waveform on Victim Interconnect

The analytic voltage waveform at the end of the victim net, that is to say, the waveform of crosstalk noise is derived in the 2-π victim wire model. In the circuit of Fig. 7.3, V_{noise} in s domain is represented as follows.

$$V_{noise}(s) = \frac{(R_{v1}R_{v2}C_{v1}s + R_{v1} + R_{v2})C_c s}{as^3 + bs^2 + ds + 1} V_{agg}(s), \tag{7.1}$$

where a, b, d are represented as follows.

$$a = R_{v1}R_{v2}R_{v3}C_{v1}(C_{v2}+C_c)C_{v3}, \tag{7.2}$$

$$b = R_{v1}C_{v1}(R_{v2}(C_{v2}+C_c+C_{v3}) + R_{v3}C_{v3}) + R_{v3}C_{v3}(C_{v2}+C_c)(R_{v1}+R_{v2}), \tag{7.3}$$

$$d = R_{v1}(C_{v1}+C_{v2}+C_c+C_{v3}) + R_{v2}(C_{v2}+C_c+C_{v3}) + R_{v3}C_{v3}. \tag{7.4}$$

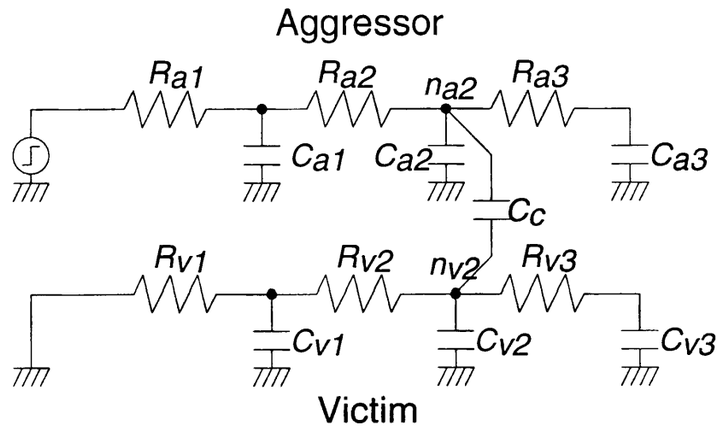


Figure 7.2: An Equivalent Circuit of Two Partially-Coupled Interconnects for Crosstalk Estimation.

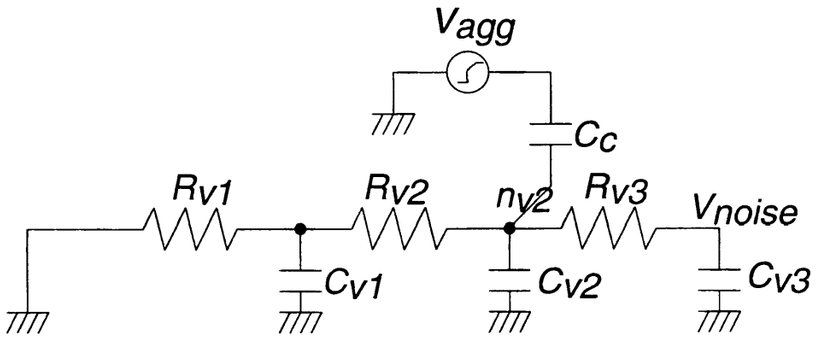


Figure 7.3: Model of Victim Wire.

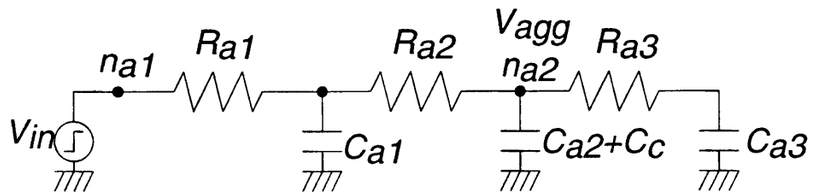


Figure 7.4: Model of Aggressive Wire.

Eq. (7.1) can be converted as follows.

$$V_{noise}(s) = \left(\frac{k_1}{s - s_1} + \frac{k_2}{s - s_2} + \frac{k_3}{s - s_3} \right) V_{agg}(s), \tag{7.5}$$

where the poles s_1, s_2, and s_3 are the roots of $as^3 + bs^2 + ds + 1 = 0$. When the relationship of $s_1 \ll s_2 \ll s_3$ is satisfied, the most dominant pole s_3 is represented as $1/d$. In this case, Eq. (7.5) can be approximated as follows.

$$V_{noise}(s) = \frac{(R_{v1} + R_{v2})C_c s}{\tau_v s + 1} V_{agg}(s), \tag{7.6}$$

where $\tau_v = d$. The voltage source of V_{agg} is assumed to be an exponential function.

$$V_{agg}(t) = V_{dd}\left(1 - e^{-t/\tau_a}\right) \quad \text{(time domain)}, \tag{7.7}$$

$$V_{agg}(s) = \frac{V_{dd}}{(\tau_a s + 1)s} \quad (s \text{ domain}). \tag{7.8}$$

Using Eq. (7.8), Eq. (7.6) is converted as follows.

$$V_{noise}(s) = \frac{(R_{v1} + R_{v2})C_c V_{dd}}{(\tau_v s + 1)(\tau_a s + 1)}. \tag{7.9}$$

The equation of the noise voltage in time domain $V_{noise}(t)$ is represented as follows.

$$V_{noise}(t) = \frac{(R_{v1} + R_{v2})C_c V_{dd}}{\tau_a - \tau_v}(e^{-\frac{t}{\tau_a}} - e^{-\frac{t}{\tau_v}}). \tag{7.10}$$

From the result of differentiating Eq. (7.10), the noise voltage becomes the peak voltage V_{peak} at the time t_{peak}.

$$V_{peak} = \frac{(R_{v1} + R_{v2})C_c V_{dd}}{\tau_v}\left(\frac{\tau_v}{\tau_a}\right)^{-\frac{\tau_a}{\tau_v - \tau_a}}, \tag{7.11}$$

$$= \frac{(R_{v1} + R_{v2})C_c V_{dd}}{\tau_a}\left(\frac{\tau_a}{\tau_v}\right)^{-\frac{\tau_v}{\tau_a - \tau_v}}, \tag{7.12}$$

$$t_{peak} = \frac{\tau_a \tau_v}{\tau_a - \tau_v} \log \frac{\tau_a}{\tau_v}. \tag{7.13}$$

7.2.2 Derivation of Aggressor Waveform

In the proposed crosstalk noise model, the aggressive signal $V_{agg}(t)$ is expressed as Eq. (7.7). Here, deriving the time constant τ_a, that is to say, the time constant at node n_{a2} in Fig. 7.4, is explained.

In Elmore delay model, the delay time between node n_{a1} and node n_{a2}, $D_{1\to2}$, is represented as follows[97].

$$D_{1\to2} = R_{a1}(C_{a1} + C_{a2} + C_c + C_{a3}) + R_{a2}(C_{a2} + C_c + C_{a3}). \tag{7.14}$$

In lumped RC networks, RC product means the transition time that a signal changes from 0% to 63%. Therefore, $D_{1\to 2}$ corresponds to the time constant at node n_{a2}, i.e. τ_a.

$$\tau_a = R_{a1}(C_{a1} + C_{a2} + C_c + C_{a3}) + R_{a2}(C_{a2} + C_c + C_{a3}). \tag{7.15}$$

The relative inaccuracy of Eq. (7.15) increases as R_{a3} becomes large compared with R_{a1} and R_{a2}. This is because the capacitance C_{a3} is shielded by the resistance R_{a3}, and the effective capacitance of C_{a3} becomes small. This effect is called "resistive shielding". In Ref. [98], a method to calculate an effective capacitance of RC networks is proposed. Using this method, the downstream network from node n_{a2} can be replaced by an effective capacitance C_{a3eff}. The effective capacitance C_{a3eff} is derived such that the amount of charge accumulated in C_{a3} and the amount of charge accumulated C_{a3eff} become the same until a time T, where T is the Elmore delay time from node n_{a1} to node n_{a2}. The effective capacitance C_{a3eff} is given by

$$C_{a3eff} = C_{a3}\left(1 - e^{-T/\tau_{dj}}\right), \tag{7.16}$$

$$T = R_{a1}(C_{a1} + C_{a2} + C_c + C_{a3}) + R_{a2}(C_{a2} + C_c + C_{a3}), \tag{7.17}$$

$$\tau_{dj} = R_{a3}C_{a3}. \tag{7.18}$$

Eq. (7.15) then becomes as follows.

$$\tau_a = R_{a1}(C_{a1} + C_{a2} + C_c + C_{a3eff}) + R_{a2}(C_{a2} + C_c + C_{a3eff}). \tag{7.19}$$

7.2.3 Driver Modeling

In the proposed crosstalk noise model, a driving CMOS gate is replaced as a resistance. The characterization of driving gates is explained. Replacing MOSFETs with resistors, a single-stage gate can be modeled as a pull-up resistance R_p, a pull-down resistor R_n, and an intrinsic output capacitance C_p(Fig. 7.5). A capacitance C_{out} is the load capacitance. MOSFETs are non-linear elements, so the value of resistance depends on the operating condition of the MOSFET. As for the aggressive wire, the output voltage swings fully between V_{DD} and V_{SS}. On the other hand, the voltage of the victim wires changes only around V_{DD} or V_{SS}. Therefore, the resistance R_p is represented as two values; the driving resistance of aggressors R_{Dp}, and the holding resistance of victims R_{Hp}. The resistance R_n is also represented as two values, R_{Dn} and R_{Hn}.

First, the driving resistance R_{Dp} is discussed. The propagating delay t_{PD}, which is the time difference between an input trip point of $0.5V_{DD}$ and output trip points of 0.37(falling, t_{PDf}) and 0.63(rising, t_{PDr}), is examined. Suppose the output signal changes low to high. The output voltage V_{out} is represented as follows.

$$V_{out}(t) = V_{DD}\left(1 - \exp^{-t/R_{Dp}(C_p + C_{out})}\right). \tag{7.20}$$

From the definition, the equation of $V_{out}(t_{PDr}) = 0.63V_{DD}$ is satisfied. The delay time t_{PDr} is hence expressed as follows.

$$t_{PDr} = R_{Dp}(C_{out} + C_p)\ln\{\frac{1}{1 - 0.63}\} \cong R_{Dp}(C_{out} + C_p). \tag{7.21}$$

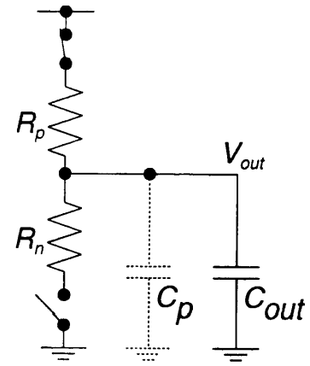


Figure 7.5: Driver Model.

The pull-up resistance R_{Dp} is determined from circuit simulation results. The delay time t_{PDr} is evaluated by circuit simulator under two conditions of C_{out}, and two sets of t_{PDr} and C_{out} are applied to Eq. (7.21), which can decide the unknown parameters R_{Dp} and C_p. Thus the pull-up resistance of aggressor R_{Dp} is characterized. The pull-down resistance R_{Dn} can be calculated similarly.

The output voltage, i.e. the noise voltage of the victim wires varies nearby V_{DD} or V_{SS}. When the noise voltage is not so large, the hold resistance R_{Hp} can be represented as the resistance in the case that the output voltage is V_{DD}. The value of the resistance R_{Hp} can be obtained by the operating condition analysis of circuit simulation. Similarly, the resistance R_{Hn} is represented as the resistance characterized in the case that the output voltage is V_{SS}.

7.2.4 Application to Generic RC Trees

In generic RC trees, many of RC trees have multiple sinks. Multiple sinks means that the tree contains branches. They also have multiple adjacent aggressive wires. Here, the method to apply generic RC trees to the 2-π victim wire model(Fig. 7.3) is discussed.

Multiple Aggressors

In linear systems, the principle of superposition holds. When the noise amplitude is not large, i.e. as long as CMOS gates can be treated as a linear resistance, the noise waveform at the sink of the victim can be represented as the superposition of the noise waveform from each aggressive wire. In this case, the maximum noise voltage at the i-th sink of the victim net, $V_{max,i}$, is represented as follows.

$$V_{max,i} = \sum_{j}^{n} V_{peak,j\to i}, \tag{7.22}$$

where n is the number of the aggressors, and $V_{peak,j\rightarrow i}$ is the noise voltage at the i-th sink caused by the j-th aggressor. The proposed method evaluates the peak noise voltage at the sink caused by each aggressors separately, and calculates the maximum noise voltage $V_{max,i}$ by Eq. (7.22).

Multiple Sinks

The noise at the i-th sink S_i caused by the j-th aggressor is considered. In this case, the trees are separated into two cases; Fig. 7.6 and Fig. 7.7. In Case 1 of Fig. 7.6, the path between the source SO and the sink S_i contains the node connected with the aggressor, n_{cc}. Conversely, in Case 2 of Fig. 7.7, the node n_{cc} is not on the path between the source SO and the sink S_i. The node n_{cc} is included within the k-th branch B_k. Reference [88] discusses the method to apply RC trees of Case 1 to the 2-π victim wire model. However the trees of Case 2 are not considered. Therefore a transformation method from the trees of Case 2 to the trees of Case 1 is devised. After this transformation, the method of Ref. [88] is applied to RC trees.

First, the method to build the 2-π victim models(Fig. 7.3) from the trees of Case 1 is explained briefly[88]. The total capacitance of the k-th branch is C_{bk}. The branch capacitances C_{bk} are added into C_{v1}, C_{v2}, and C_{v3} in Fig. 7.3 in the following manner:

- When a branch B_k is between SO and n_{cc}, the resistance between SO and n_k, R_{SO-n_k}, is represented as $R_{SO-n_k} = \alpha \cdot R_{SO-n_{cc}}$, where $0 \leq \alpha \leq 1$. Then $\alpha \cdot C_{bk}$ is added to C_{v2}, and $(1-\alpha) \cdot C_{bk}$ is added to C_{v1}.

- When a branch B_k is between n_{cc} and S_i, the resistance between n_{cc} and S_i, $R_{n_{cc}-S_i}$, is represented as $R_{n_k-Si} = \beta \cdot R_{n_{cc}-S_i}$, where $0 \leq \beta \leq 1$. Then $\beta \cdot C_{bk}$ is added to C_{v2}, and $(1-\beta) \cdot C_{bk}$ is added to C_{v3}.

Next, the transformation method from Case 2 to Case 1 is explained. At first, the coupling capacitance is moved from the node n_{cc} to the node n_k(Fig. 7.7). This simple movement, however, may cause the overestimation of noise voltage. Though the amount of the influence from the aggressor is decreased by the resistance between n_k and n_{cc}, this degradation is not considered at all. The proposed transformation method treats this degradation as the increase in the time constant of the voltage source τ_a. The Elmore delay from n_{cc} to n_k is added to τ_a. Finally, the capacitance of the branch B_k is connected to n_k, i.e. C_{bk} is added to C_{v2}. By the above procedure, the trees of Case 2 are converted to the trees of Case1. Afterward, 2-π models are obtained by the method of Ref. [88].

7.3 Optimization Algorithm

From the discussion in the previous section, crosstalk noise can be estimated for any interconnects in a circuit. In this section, the optimization algorithm for crosstalk noise reduction is discussed. The proposed algorithm reduces crosstalk noise under delay and transition time

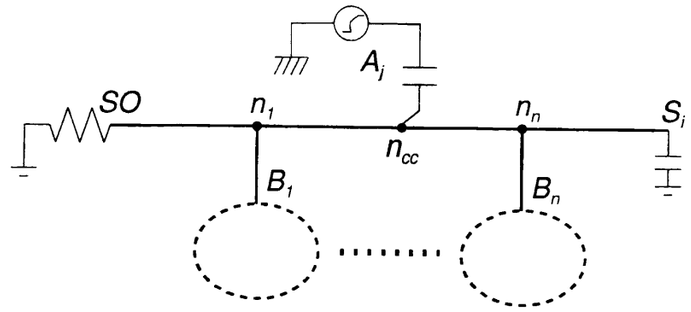


Figure 7.6: An Interconnect with Branches(Case 1).

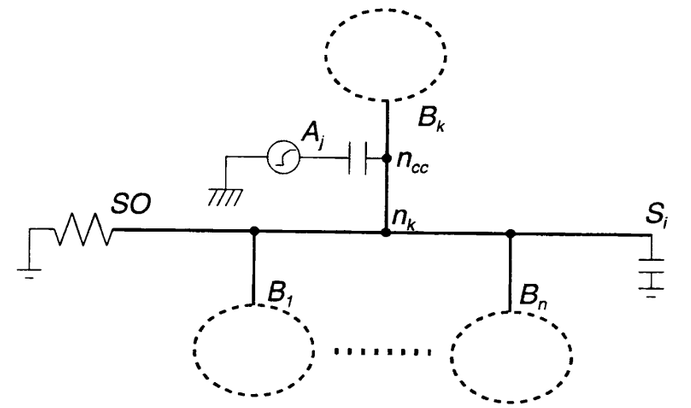


Figure 7.7: An Interconnect with Branches(Case 2).

constraints. First, the optimization algorithm for the localized problem that includes one victim net and its adjacent nets is explained. This section then shows the overall algorithm that builds and solves the local optimization problems, considering the global optimality under delay constraints.

7.3.1 Optimization Algorithm in Each Victim Net

First, the noise reduction algorithm for each victim net is explained. The proposed method downsizes the drivers of the adjacent aggressive wires in order to reduce the amount of crosstalk noise at the victim wire. When the driving strength of the aggressive wire becomes weak, i.e. the driver resistance R_{a1} becomes large, the time constant of the aggressive voltage source τ_a increases(Eq. (7.15)). From Eq. (7.12), the maximum noise voltage V_{peak} at the victim net consequently decreases.

In order to choose the driver of the aggressive wire to be downsized efficiently, a measure

$priority$ is devised.

$$priority_i = slack_i \cdot \sum_j^n V_{peak,i \to j}, \tag{7.23}$$

where $V_{peak,i \to j}$ is the noise voltage at the j-th sink caused by the i-th aggressive net, and n is the number of sinks. The value $slack_i$ represents the timing margin at the i-th aggressive net, and is defined as the time difference between the required time and the arrival time[40]. The measure $priority_i$ becomes large in the case that the i-th adjacent net causes a large amount of noise and the timing constraint at the i-th aggressive net is not tight. Using this measure, the proposed algorithm can find the aggressive net efficiently that has strong influence on the crosstalk noise at the victim net yet has little influence on the circuit delay.

Step 1: Calculate $priority$(Eq. (7.23)) for each adjacent aggressive net, and put all the aggressive nets into list L_l.

Step 2: Choose the aggressive net with the maximum $priority$ from list L_l.

Step 3: Downsize the driver of the chosen aggressive net within the limit that the delay constraints and the transition time constraints are satisfied. The best size of the driver is decided such that the value of $(V_v^2 + V_a^2)$ becomes the smallest, where V_v is the noise voltage at the victim net, and V_a is the noise voltage at the aggressive net. Remove the aggressive net from L_l.

Step 4: If the noise voltage becomes smaller than the target value V_{target}, or if the list L_l becomes empty, finish the optimization procedure. Otherwise go back to **Step 2**. The value V_{target} is explained in the following section.

7.3.2 Overall Optimization Algorithm

Section 7.3.1 discusses the optimization algorithm for the localized problem that contains one victim net and its adjacent aggressive nets. Next, the overall algorithm is discussed. This algorithm aims to reduce both the maximum noise voltage in a circuit and the number of nets with large amounts of noise.

The optimization iterates the following procedure from **Stage 1** to **Stage 4** for several times, as the value $threshold$ is gradually decreased. The parameter $threshold$ is used for selecting the nets to be optimized, and it ranges from 0 to 1. The nets whose noise voltages are larger than the product of $threshold$ and the maximum noise voltage in the circuit are chosen as the optimization candidates. In the beginning, $threshold$ is set close to 1 in order to reduce the maximum noise voltage intensively. In the end, $threshold$ is set close to 0, and the most of the nets in the circuit are optimized.

Stage 1: Calculate the crosstalk noise at each net in the circuit.

Stage 2: Find the maximum voltage of crosstalk noise V_{max} in the circuit, and put the nets whose noise voltages are larger than $V_{max} \times threshold$ into the candidate list L_o.

Stage 3: Choose the net with the maximum noise voltage in the list L_o, and execute the optimization explained in Sec. 7.3.1. The value of $V_{max} \times threshold$ is given to the optimization as the target value. Remove the net from the list L_o.

Stage 4: If the list L_o becomes empty, finish the optimization procedure. Otherwise go back to **Stage 3**.

When the timing constraints are given, the timing margin at each net should be utilized efficiently for reducing the crosstalk noise. Therefore the sequence of the nets to be optimized is critical and essential to obtain high-quality circuits. In order to reduce the maximum noise voltage, the proposed algorithm gives priority to the net with large noise. **Stage 2** excludes the nets whose noise voltages are smaller than $V_{max} \times threshold$ from the the optimization candidates. In **Stage 3**, the nets are optimized in order of the amount of noise voltage.

In **Stage 3**, the target noise value $V_{max} \times threshold$ is given to the localized optimization problem, in order to control the local optimization from the viewpoint of global optimality. The optimization result that the noise voltage is minimized in the localized problem may incur a bad local-minimum solution globally. This is because the timing margins, which may be utilized for reducing the noise at other nets, are wasted. The proposed algorithm hence stops the local optimization when the noise voltage becomes smaller than the target value. Thanks to the good sequence of the net to be optimized and setting the target noise value, the proposed method can reach a good solution under the delay constraints.

7.4 Experimental Results

This section shows some experimental results. First the accuracy of the crosstalk noise model is demonstrated. Next the optimization results for crosstalk noise reduction are shown.

The circuits used for the experiments are an ALU in a DSP for mobile phone[67] (dsp_alu) and the circuits included LGSynth93 benchmark sets (des). These circuits are synthesized for minimizing the circuit delay[56]. The circuit scale of dsp_alu is 12547 cells, and the number of cells in des is 3414. The layouts of the synthesized circuits are generated. The layout area of dsp_alu is 5.3(2.3x2.3)mm^2, and the area of des is 0.64(0.8x0.8)mm^2. RC trees of interconnects are extracted from the layouts by a quasi-3D RC extract tool[99]. The coupling capacitances below 10fF are extracted as the capacitance to the ground, where the coupling capacitance of 10fF corresponds to the length of 230 μm. The supply voltage is 3.3V.

Cell layouts are generated using VARDS[27] in a 0.35μm process with three metal layers. The layout generation system VARDS can vary transistor widths in a cell while keeping the location of each pin. Exploiting this feature, the proposed method optimizes a detail-routed circuit without any wire modifications. The height of the generated cells is 13 interconnect-pitches, and the size ratio of PMOS and NMOS transistors is 1. In transistor sizing, MOSFETs are down-sized within the range that VARDS can generate cell layouts. The maximum transistor width of standard driving-strength(x1) cells is 6.2μm. The transistor width can be reduced to 0.9μm.

7.4.1 Crosstalk Estimation

The accuracy of the crosstalk noise model is discussed. First, the peak voltage of the crosstalk noise is evaluated using the model circuit shown in Fig. 7.2. In this model circuit, the appropriateness of the following three points can be experimentally verified; the separation into two circuits of Fig. 7.3 and Fig. 7.4, the approximation used in Eq. (7.6), and the derivation of the time constant τ_a in Eq. (7.19). The peak voltage of the crosstalk noise is evaluated by circuit simulation, the conventional method[88], and the proposed method. In the conventional method[88], the signal from the aggressive wire $V_{agg(t)}$ is represented as a saturated lump function.

$$V_{agg}(t) = \begin{cases} \frac{t}{t_r} \cdot V_{DD} & (0 \leq t \leq t_r), \\ V_{DD} & (t \geq t_r). \end{cases} \tag{7.24}$$

However the calculation method of t_r is not explained. In this experiment, the transition time t_r is calculated as $\tau_a \times 2.7$. The coefficient of 2.7 is determined such that the sum of the absolute error between the simulation results and the results estimated by Ref. [88] is minimized. The parameters extracted from the actual RC trees in the layout of des are utilized as $R_{a1}, R_{a2}, R_{a3}, C_{a1}, C_{a2}, C_{a3}, R_{v1}, R_{v2}, R_{v3}, C_{v1}, C_{v2}, C_{v3}$. The peak noise is evaluated for all the coupled interconnects in des circuit. Fig. 7.8 shows the estimation results by the proposed method. The horizontal axis is the noise voltage evaluated by circuit simulation and the vertical axis is the voltage estimated by the proposed method. The diagonal line represents the ideal line with 0 error. The proposed method estimates the peak noise voltage accurately. The average estimation error is 1.6%. Fig. 7.9 represents the result by the conventional method. Compare with the proposed method, the estimation accuracy is not high. The average error of the conventional method is 28.1%. By adopting an exponential function as

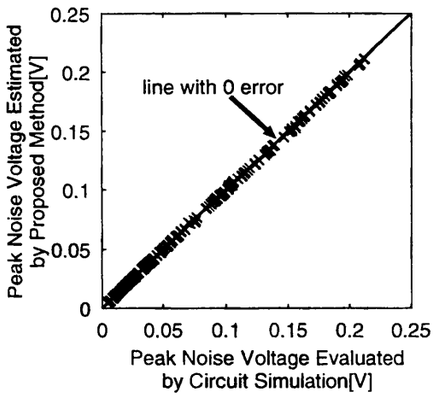


Figure 7.8: Peak Noise Estimation in Fig. 7.2 Model by Proposed Method(des).

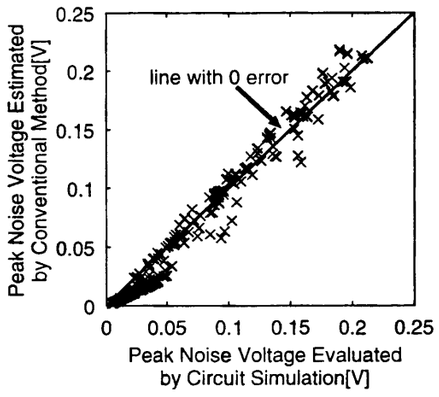


Figure 7.9: Peak Noise Estimation in Fig. 7.2 Model by Conventional Method[88] (des).

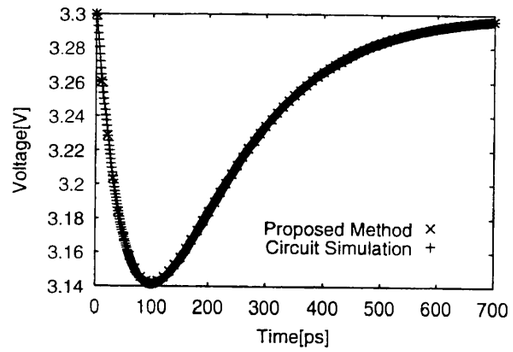


Figure 7.10: An Example of the Crosstalk Noise Waveform.

the signal waveform from the aggressor, the estimation accuracy improves. Fig. 7.10 shows an example of the waveforms evaluated by circuit simulation and the proposed method. The waveform of the crosstalk noise is estimated precisely by the proposed method.

Next, the crosstalk noise is evaluated in more actual circuits, i.e. the drivers and the receivers are CMOS gates and the RC trees have branches. The circuits used for this experiment are included in des circuit. Fig. 7.11 shows the estimation results of peak crosstalk noise. The average error of the maximum noise estimation is 22.3% and 10mV in the proposed method. In order to examine the effectiveness of the transformation method from Case 2 to Case 1 discussed in Sec. 7.2.4, the crosstalk noise is evaluated by the following simple method. The coupling capacitance is moved from the node n_{cc} to the node n_k, and the capacitance of the branch B_k is connected to n_k (Fig. 7.7). This simple method does not adjust the time constant of the aggressive signal τ_a, which is the difference between the proposed method and this simple method. Fig. 7.12 shows the estimation results by the simple method. There is not a significant difference between Fig. 7.11 and Fig. 7.12. The average error of the simple method is 22.6%, which is only 0.3% larger than the proposed method. This is because the interconnect resistance is not high in the 0.35μm technology, and hence the resistance between n_k and n_{cc} scarcely affects the crosstalk noise. Therefore, the crosstalk noise is evaluated in the circuit of Fig. 7.13 by circuit simulation, the simple method, and the proposed method, assuming a 0.13μm technology. The values of resistance and capacitance are calculated under the interconnect structure shown in Table 1.1. The driver resistances of the victim and the aggressor is 1kΩ, and the input capacitance of the receivers is 10fF. The variable x represents the distance between the junction and the start point of coupling. Fig. 7.14 shows the results. The peak noise voltage decreases as the distance x increases. The proposed method shows the tendency for noise to decrease, whereas the noise voltage estimated by the simple method is constant. However, the shape of decrease is different from the simulation results. Further improvement is required.

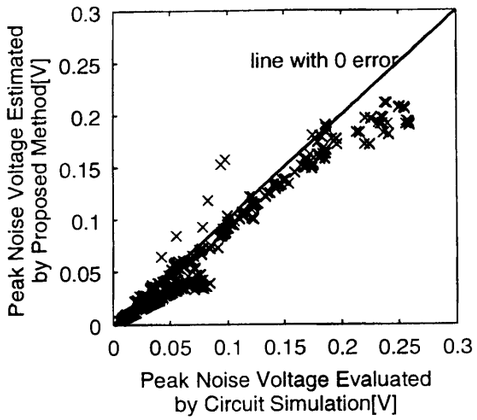


Figure 7.11: Peak Noise Estimation with CMOS Gates and Branch Trees by Proposed Method(des).

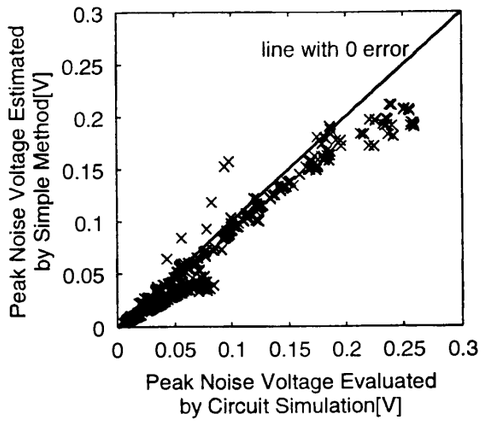


Figure 7.12: Peak Noise Estimation with CMOS Gates and Branch Trees by Simple Method(des).

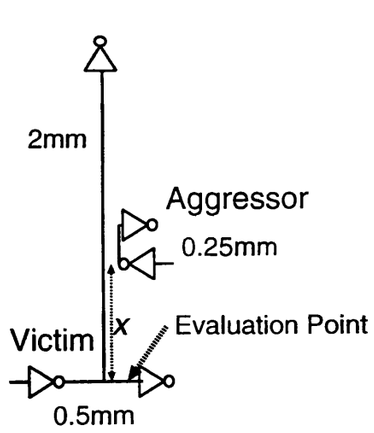


Figure 7.13: Interconnect Structure used for Crosstalk Noise Evaluation.

7.4.2 Crosstalk Reduction

The optimization results for crosstalk noise reduction are shown. The circuits are optimized under the delay constraints of the initial circuits' delay time. The given constraint of the transition time is 1.0ns. Figs. 7.15 and 7.16 show the distributions of the maximum noise voltage before and after the optimization. In des circuit, the maximum noise voltage is reduced from 0.40V to 0.20V by 50%. The distribution is also shifted in the direction that the noise voltage decreases. In dsp_alu circuit, the maximum noise is reduced from 0.99V to 0.62V by 37%. The number of nets whose noise voltages are over 0.5V is decreased from

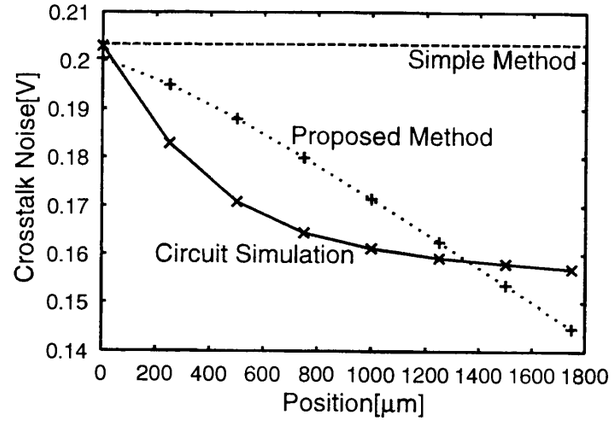


Figure 7.14: Peak Noise Evaluation in the Circuit of Fig. 7.13.

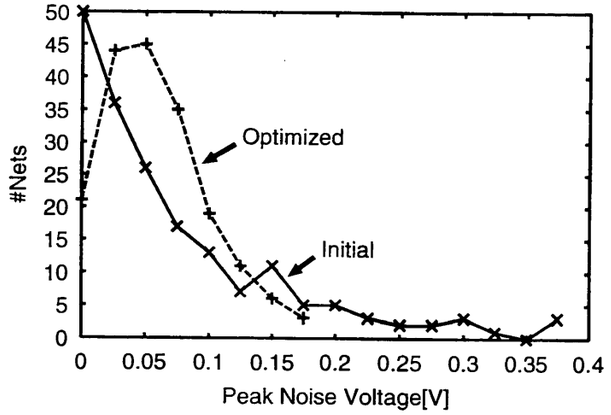


Figure 7.15: Optimization Results for Crosstalk Noise Reduction (des).

109 to 4. The CPU times required for the optimization on an Alpha Station are 111 seconds in des(3.4k cells), and 6726 seconds in dsp_alu(13k cells). After the detailed-routing, the crosstalk noise can be reduced considerably by only downsizing the transistors inside cells while preserving the interconnects. The circuit delay is also preserved.

7.5 Conclusion

This chapter proposes an optimization method for crosstalk noise reduction by transistor sizing. The proposed method optimizes the detail-routed circuits such that MOSFETs inside

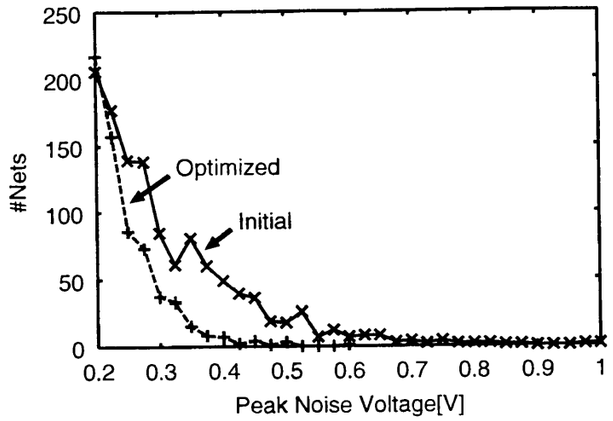


Figure 7.16: Optimization Results for Crosstalk Noise Reduction (dsp_alu).

cells are downsized without any interconnect modifications, based on the crosstalk noise estimation by analytic noise expressions. The effectiveness of the proposed method is experimentally verified using 2 benchmark circuits. The maximum noise voltage is reduced by more than 35% without delay increase, which contributes to high-reliability LSI design.

Chapter 8

Conclusion

This thesis discusses performance optimization techniques in physical design. In DSM technology, interconnect delay, power dissipation, delay fluctuation and crosstalk noise become the severe problems that limit, or rather deteriorate the circuit performance. Reducing interconnect delay is intensively studied, and effective solutions are developed. Compared with interconnect delay, other problems are not sufficiently considered. This thesis focuses on power dissipation, delay fluctuation and crosstalk noise, and proposes solutions in physical design for each problem. The proposed techniques are expected to be more essential and contribute to design high-performance and high-reliability LSIs in future technology, since these problems originate in shrinking feature size.

In Chapter 2, a performance optimization method by input reordering is discussed. The input pins, whose logical functions are the same though, in a cell have the different characteristics in delay and power dissipation, which is utilized for delay and power reduction. The effectiveness of the proposed method is experimentally examined using 30 benchmark circuits. Power dissipation is reduced by 22.5% maximum and by 5.9% on average. The proposed method also reduces delay time by 6.7%. It is verified that input reordering improves circuit performance steadily with almost zero penalty.

Chapter 3 discusses a gate sizing method that reduces glitch power dissipation. A statistical glitch estimation method and a gate sizing algorithm that explores solution space globally are developed. Thanks to them, the proposed method can optimize the number of glitches as well as capacitive load and short circuit power dissipation, whereas conventional methods assume the number of glitches to be constant. Power dissipation is reduced by 16.2% maximum and by 10.4% on average further from the minimum-area circuits, where the conventional methods consider the minimum-area circuits as the minimum-power circuits.

Chapter 4 discusses a performance optimization method based on statistical timing analysis. This method aims to remove both over-design and under-design for high-performance and high-reliable LSI design. The proposed method focuses on the local delay fluctuation, and calculates the statistically-distributed circuit delay. Slack, which represents the timing criticality at each cell under a deterministic delay model and is widely used for performance optimization, can not be defined under the statistical delay model. Therefore a new measure of timing criticality for statistical delay model is devised, and the optimization algorithm us-

ing this measure is developed. The worst-case delay can be estimated within 3% error by the statistical timing analysis method. It is verified that the proposed method can reduce delay and power dissipation from the circuits optimized without considering delay fluctuation.

Chapter 6 discusses that performance optimization involves undesirable secondary effect that the optimized circuits become sensitive to delay uncertainty. Some examples of the increase in delay uncertainty are demonstrated, and this chapter cautions that performance optimization may cause an involuntary delay violation. It is also verified that the statistical timing analysis discussed in Chapter 4 is effective as one of solutions of this problem.

Chapter 5 and Chapter 7 show the performance optimization methods based on a design framework that can vary transistor sizes inside a cell flexibly without any interconnect modifications. This framework aims to design a circuit whose performance is close to that of full-custom design, making the best use of usual cell-base design tools. Chapter 5 discusses a power reduction method that downsizes transistors after detail-routing. Power dissipation can be reduced as much as possible without delay violation, since the optimized layout can be obtained preserving interconnects. The proposed method reduces power dissipation by 77% maximum and 65% on average without any delay increase from the cell-based circuits. This method also contributes to increase the reliability of the circuits by reducing current density.

In Chapter 7, a transistor sizing method for reducing crosstalk noise is discussed. This method optimizes the detail-routed circuits, estimating crosstalk noise based on the interconnect information extracted from the layout. The conventional circuit optimization techniques involves a certain amount of wiring variation when the optimization result is applied to the layout, which makes it difficult to optimize crosstalk noise by circuit optimization. However, the proposed method can reduce crosstalk noise efficiently, because the proposed method can vary transistor sizes inside cells without interconnect modifications. The analytic expressions of peak crosstalk noise are derived and used for crosstalk noise estimation. The optimization algorithm for crosstalk noise reduction that can consider delay constraints well is developed. Utilizing them, the crosstalk noise is reduced by downsizing the drivers of the aggressive wires. The effectiveness of the proposed method is examined using 2 circuits. The maximum noise voltage can be reduced by more than 35% without any delay increase.

The future work includes constructing an overall design methodology that optimizes all metrics of circuit performance simultaneously, such as delay, power dissipation, and area, considering delay uncertainty and crosstalk noise.

Bibliography

[1] J. P. Fishburn and A. E. Dunlop, "Tilos: A Posynomial Programming Approach to Transistor Sizing," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 326–328, 1985.

[2] M. R. C. M. Berkelaar and J. A. G. Jess," Gate Sizing in MOS Digital Circuits with Linear Programming," In *Proceedings of European Design Automation Conference*, pp.217-221, 1990.

[3] S. S. Sapatnekar, V. B. Rao, P. M. Vaidya and S. M. Kang, "An Exact Solution to the Transistor Sizing Problem for CMOS Circuits using Convex Optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 12, pp.1621-1634, Nov. 1993.

[4] Y. Tamiya and Y. Matsunaga, "LP Based Cell Selection with Constraints of Timing, Area, and Power Consumption," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 378–381, 1994.

[5] G. Chen, H. Onodera and K. Tamaru," An Iterative Gate Sizing Approach with Accurate Delay Evaluation," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp.422-427, 1995.

[6] K. Sato, M. Kawarabayashi, H. Emura, and N. Maeda, "Post-Layout Optimization for Deep Submicron Design," In *Proceedings of IEEE/ACM Design Automation Conference*, pp. 740-745, 1996.

[7] Y. Jiang, S. S. Sapatnekar, C. Bamji, and J. Kim, "Interleaving Buffer Insertion and Transistor Sizing into a Single Optimization," *IEEE Transactions on Very Large Scale Integration(VLSI) Systems*, Vol. 6, No. 4, pp.625-633, Dec. 1998.

[8] T. Okamoto and J. Cong, "Buffered Steiner Tree Construction with Wire Sizing for Interconnect Layout Optimization," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp.44-49, 1996.

[9] C. Alpert and A. Devgan, "Wire Segmenting for Improved Buffer Insertion," In *Proceedings of IEEE/ACM Design Automation Conference*, pp.588-593, 1997.

[10] T. Sakurai, "Closed-Form Expressions for Interconnection Delay, Coupling, and Crosstalk in VLSI's," *IEEE Transactions on Electron Devices*, Vol. 40, No. 1, January 1993.

[11] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, 1999.

[12] J. Culetu, C. Amir, and J. MacDonald, "A Practical Repeater Insertion Method in High Speed VLSI Circuits," In *Proceedings of IEEE/ACM Design Automation Conference*, pp.392-395, 1998.

[13] J. Lillis and C.-K. Cheng, "Timing Optimization for Multisource Nets: Characterization and Optimal Repeater Insertion," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 18, No. 3, March 1999.

[14] D. Li, A. Pua, P. Srivastava, and U. Ko, "A Repeater Optimization Methodology for Deep Sub-Micron, High-Performance Processors," In *Proceedings of IEEE International Conference on Computer Design*, pp.726-731, 1997.

[15] N. Kojima, Y. Parameswar, C. Klingner, Y. Ohtaguro, M. Matsui, S. Iwasa, T. Teruyama, T. Shimazawa, H. Takeda, K. Hashizume, H. Tago, and M. Yamada, "Repeater Insertion Method and its Applications to a 300MHz 128-bit 2-way Superscalar Microprocessor," In *Proceedings of Asia and South Pacific Design Automation Conference*, pp.641-646, 2000.

[16] I-M. Liu, A. Aziz, and D. F. Wong, "Meeting Delay Constraints in DSM by Minimal Repeater Insertion," In *Proceedings of Design Automation and Test in Europe*, pp.436-440, 2000.

[17] J. Cong and C.-K. Koh, "Simultaneous Driver and Wire Sizing for Performance and Power Optimization," *IEEE Transactions on Very Large Scale Integration(VLSI) Systems*, Vol. 2, No. 4, pp.408-425, December 1994.

[18] S. S. Sapatnekar, "Wire Sizing as a Convex Optimization Problem: Exploring the Area-Delay Tradeoff," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 15, No. 8, pp.1001-1011, August 1996.

[19] N. Menezes, R. Baldick, and L. T. Pileggi, "A Sequential Quadratic Programming Approach to Concurrent Gate and Wire Sizing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 16, No. 8, pp.867-881, August 1997.

[20] C.-P. Chen, C. C. N. Chu, and D. F. Wong, "Fast and Exact Simultaneous Gate and Wire Sizing by Lagrangian Relaxation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 18, No. 7, July 1999.

[21] C. C. N. Chu and D. F. Wong, "An Efficient and Optimal Algorithm for Simultaneous Buffer and Wire Sizing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 18, No. 9, September 1999.

[22] C. C. N. Chu and D. F. Wong, "A Quadratic Programming Approach to Simultaneous Buffer Insertion/Sizing and Wire Sizing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 18, No. 6, June 1999.

[23] J. Lillis, C.-K. Cheng, and T.-T. Y. Lin, "Optimal Wire Sizing and Buffer Insertion for Low Power and a Generalized Delay Model," *IEEE Journal of Solid-State Circuits*, Vol. 31, No. 3, March 1996.

[24] S. Nassif, "Delay Variability: Sources, Impacts and Trends," In *Proceedings of IEEE International Solid-State Circuits Conference*, pp.368-369, 2000.

[25] M. Berkelaar and E. Jacobs, "Sources and Quantification of Delay Variations in a 250nm CMOS Digital Cell Library," In *Proceedings of International Workshop on Logic Synthesis*, pp. 335-339, 2000.

[26] D. Sylvester and K. Keutzer, "Getting to the Bottom of Deep Submicron," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp.203-211, 1998.

[27] T. Hashimoto and H. Onodera, "Layout Generation of Primitive Cells with Variable Driving Strength," In *Proceedings of the Ninth Workshop on Synthesis and System Integration of Mixed Technologies*, pp.122-129, 2000.

[28] K. L. Shepard, S. M. Carey, E. K. Cho, B. W. Curran, R. F. Hatch, D. E. Hoffman, S. A. McCabe, G. A. Northrop, and R. Seigler, "Design Methodology for the S/390 Parallel Enterprise Server G4 Microprocessors," *IBM Journal of Research and Development*, Vol. 41, No. 4/5, pp. 515-547, September 1997.

[29] A. P. Chandrakasan, S. Sheng and R. W. Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 4, pp.473-484, April 1992.

[30] K. Usami and M. Horowitz, "Clustered Voltage Scaling Technique for Low-Power Design," In *Proceedings of IEEE/ACM International Symposium on Low Power Design*, pp.3-8, 1995.

[31] C.-Y. Tsui, M. Pedram and A. M. Despain, "Power Efficient Technology Decomposition and Mapping under an Extended Power Consumption Model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 13, No. 9, pp.1110-1122, September 1994.

[32] M. Hashimoto, H. Onodera and K. Tamaru, "A Practical Gate Resizing Technique Considering Glitch Reduction for Low Power Design," In *Proceedings of IEEE/ACM Design Automation Conference*, pp.446-451, 1999.

[33] B. Lin and H. De Man, "Low-Power Driven Technology Mapping under Timing Constraints," In *Proceedings of IEEE International Conference on Computer Design*, pp.421-427, 1993.

[34] W.-Z. Shen, J.-Y. Lin, and F.-W. Wang, "Transistor Reordering Rules for Power Reduction in CMOS Gates," In *Proceedings of Asia and South Pacific Design Automation Conference*, pp.1-6, 1995.

[35] R. Hossain, M. Zheng and A. Albicki, "Reducing Power Dissipation in CMOS Circuits by Signal Probability Based Transistor Reordering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 15, No. 3, pp.361-368, March 1996.

[36] E. Musoll and J. Cortadella, "Optimizing CMOS Circuits for Low Power using Transistor Reordering," In *Proceedings of European Design and Test Conference*, pp.219-223, 1996.

[37] S. C. Prasad and K. Roy, "Transistor Reordering for Power Minimization under Delay Constraint," *ACM Transactions on Design Automation of Electronic Systems*, Vol. 1, No. 2, pp.280-300, April 1996.

[38] B. S. Carlson and S.-J. Lee, "Delay Optimization of Digital CMOS VLSI Circuits by Transistor Reordering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 14, No. 10, pp.1183-1192, October 1995.

[39] M. Marek-Sadowska and S. P. Lin, "Pin Assignment for Improved Performance in Standard Cell Design," In *Proceedings of IEEE International Conference on Computer Design*, pp.339-342, 1990.

[40] R. B. Hitchcock, G. L. Smith and D. D. Cheng, "Timing Analysis of Computer Hardware," *IBM Journal of Research and Development*, Vol. 26, No. 1, pp.100-105, January 1982.

[41] PowerMill Reference Manual. Synopsys, Inc., CA, 1999.

[42] F. N. Najm, "Transition Density, a Stochastic Measure of Activity in Digital Circuits," In *Proceedings of IEEE/ACM Design Automation Conference*, pp.644-649, 1991.

[43] D. Brand and C. Visweswariah, "Inaccuracies in Power Estimation during Logic Synthesis," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp.388-394, 1996.

[44] H. Onodera, A. Hirata, T. Kitamura and K. Tamaru, "P2lib : Process-Portable Library and Its Generation System," In *Proceedings of IEEE Custom Integrated Circuits Conference*, pp.341-344, 1997.

[45] M. Hashimoto, H. Onodera, and K. Tamaru, "A Power and Delay Optimization Method using Input Reordering in Cell-Based CMOS Circuits," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science*, Vol. E82-A, No. 1, pp. 159-166, 1999.

[46] G. Chen, H. Onodera, and K. Tamaru, "Timing and Power Optimization by Gate Sizing Considering False Path," In *Proceedings of the Sixth Great Lakes Symposium on VLSI*, pp. 154–159, 1996.

[47] M. Borah, R. M. Owens, and M. J. Irwin, "Transistor Sizing for Minimizing Power Consumption of CMOS Circuits under Delay Constraint," In *Proceedings of International Symposium on Low Power Design*, pp. 167–172, 1995.

[48] H.-R. Lin and T. T. Hwang, "Power Reduction by Gate Sizing with Path-Oriented Slack Calculation," In *Proceedings of Asia-Pacific Design Automation Conference*, pp. 7–12, 1995.

[49] S. Devadas A. Shen, A. Ghosh and K. Keutzer, "On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 402–407, 1992.

[50] F. Najm and M. Y. Zhang, "Extreme Delay Sensitivity and the Worst-Case Switching Activity in VLSI Circuits," In *Proceedings of IEEE/ACM Design Automation Conference*, pp. 623–627, 1995.

[51] S. S. Sapatnekar and W. Chuang, "Power vs. Delay in Gate Sizing: Conflicting Objectives?," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 463–466, 1995.

[52] V. D. Agrawal, M. L. Bushnell, G. Parthasarathy and R. Ramadoss, "Digital Circuit Design for Minimum Transient Energy and a Linear Programming Method," In *Proceedings of International Conference on VLSI Design*, pp. 434–439, 1999.

[53] L. Benini, G. D. Micheli, A. Macii, E. Macii, M. Poncino and R. Scarsi, "Glitch Power Minimization by Selective Gate Freezing," *IEEE Transactions on Very Large Scale Integration(VLSI) Systems*, Vol. 8, No. 3, pp.287-298, June 2000.

[54] Y. J. Lim and M. Soma, "Statistical Estimation of Delay-Dependent Switching Activities in Embedded CMOS Combinational Circuits," *IEEE Transaction on Very Large Scale Integration(VLSI) Systems*, Vol. 5, No. 3, pp. 309–319, September 1997.

[55] M. Berkelaar, "Statistical Delay Calculation, a Linear Time Method," In *Proceedings of IEEE/ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pp. 15–24, 1997.

[56] Synopsys Inc., *Design Compiler Reference Manual*, 1998.

[57] H.-F. Jyu, S. Malik, S. Devadas and K. W. Keutzer, "Statistical Timing Analysis of Combinational Logic Circuits," *IEEE Transactions on Very Large Scale Integration(VLSI) Systems*, Vol. 1, No. 2, pp.126-137, June 1993.

[58] R.-B. Lin and M.-C. Wu, "A New Statistical Approach to Timing Analysis of VLSI Circuits," In *Proceedings of International Conference on VLSI Design*, pp.507-513, 1997.

[59] R. B. Brashear, N. Menezes, C. Oh, L. T. Pillage and M. R. Mercer, "Predicting Circuit Performance Using Circuit-level Statistical Timing Analysis," In *Proceedings of European Design and Test Conference*, pp.332-337, 1994.

[60] E.T.A.F. Jacobs and M.R.C.M. Berkelaar, "Gate Sizing Using a Statistical Delay Model," In *Proceedings of Design Automation and Test in Europe*, pp.283-290, 2000.

[61] O. Coudert, "Gate Sizing for Constrained Delay/Power/Area Optimization," *IEEE Transactions on Very Large Scale Integration(VLSI) Systems*, Vol. 5, No. 4, pp. 465-472, December 1997.

[62] D. -S. Chen and M. Sarrafzadeh, "An Exact Algorithm for Low Power Library-Specific Gate Re-Sizing," In *Proceedings of IEEE/ACM Design Automation Conference*, pp. 783-788, 1996.

[63] M. Orshansky, C. Spanos and C. Hu, "Circuit Performance Variability Decomposition," In *Proceedings of International Workshop on Statistical Metrology*, pp. 10-13, 1999.

[64] M. Yamada, S. Kurosawa, R. Nojima, N. Kojima, T. Mitsuhashi, and N. Goto, "Synergistic Power/Area Optimization with Transistor Sizing and Wire Length Minimization," *IEICE Transactions on Electrons*, Vol. E78-C, No. 4, pp. 441-446, 1995.

[65] J. P. Uyemura, "Funcamentals of MOS Digital Integrated Circuits," Addison-Wesley Publishing Company, 1988.

[66] H. Y. Chen and S. M. Kang, "A New Circuit Optimization Technique for High Performance CMOS Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 10, No. 5, May 1991.

[67] T. Iwahashi, T. Shibayama, M. Hashimoto, K. Kobayashi and H. Onodera, "Vector Quantization Processor for Mobile Video Communication," In *Proceedings of IEEE International ASIC/SOC Conference*, pp.75-79, 2000.

[68] PathMill Reference Manual. Synopsys, Inc., CA, 1999.

[69] P. Yang and J.-H. Chern, "Design for Reliability: The Major Challenge for VLSI," In *Proceedings of IEEE*, Vol. 81, No. 5, May 1993.

[70] M. Hashimoto and H. Onodera, "A Performance Optimization Method by Gate Resizing Based on Statistical Static Timing Analysis," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E83-A, No. 12, pp. 2558-2568, December 2000.

[71] M. Hashimoto and H. Onodera, "Post-Layout Transistor Sizing for Power Reduction in Cell-Based Design," In *Proceedings of Asia and South Pacific Design Automation Conference*, pp. 359-365, 2001.

[72] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V Power Supply High-Speed Digital Circuit Technology with Multithreshold-Voltage CMOS," *IEEE Journal of Solid-Stage Circuits*, Vol. 30, No. 8, August 1995.

[73] M. Ketkar, K. Kasamsetty, and S. Sapatnekar, "Convex Delay Models for Transistor Sizing," In *Proceedings of IEEE/ACM Design Automation Conference*, pp.655–660, 2000.

[74] K. Chaudhary, A. Onozawa, and E. S. Kuh, "A Spacing Algorithm for Performance Enhancement and Cross-talk Reduction," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp.697-702, 1993.

[75] T. Gao and C. L. Liu, "Minimum Crosstalk Switchbox Routing," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp.610-615, 1994.

[76] D. A. Kirkpatrick and A. L. Sangiovanni-Vincentelli, "Techniques for Crosstalk Avoidance in the Physical Design of High-Performance Digital Systems," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp.616-619, 1994.

[77] H. Zhou and D. F. Wong, "An Optimal Algorithm for River Routing with Crosstalk Constraints," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp.310-315, 1996.

[78] H. Zhou, and D. F. Wong, "Global Routing with Crosstalk Constraints," In *Proceedings of IEEE/ACM Design Automation Conference*, pp.374-377, 1998.

[79] H.-P. Tseng, L. Scheffer, and C. Sechen, "Timing and Crosstalk Driven Area Routing," In *Proceedings of IEEE/ACM Design Automation Conference*, pp.378-381, 1998.

[80] P. Saxena, and C. L. Liu, "Crosstalk Minimization using Wire Perturbations," In *Proceedings of IEEE/ACM Design Automation Conference*, pp.100-103, 1999.

[81] T. Xue, E. S. Kuh, and D. Wang, "Post Global Routing Crosstalk Risk Estimation and Reduction," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp.302-309, 1996.

[82] C.-C. Chang, and J. Cong, "Pseudo Pin Assignment with Crosstalk Noise Control," In *Proceedings of ACM International Symposium on Physical Design*, pp.41-47, 2000.

[83] R. Kay, and R. A. Rutenbar, "Wire Packing: A Strong Formulation of Crosstalk-Aware Chip-Level Track/Layer Assignment with an Efficient Integer Programming Solution," In *Proceedings of ACM International Symposium on Physical Design*, pp.61-68, 2000.

[84] I. H.-R. Jiang, S.-R. Pan, Y.-W. Chang, and J.-Y. Jou, "Optimal Reliable Crosstalk-Driven Interconnect Optimization," In *Proceedings of ACM International Symposium on Physical Design*, pp.128-133, 2000.

[85] C.-P. Chen and N. Menezes, "Noise-aware Repeater Insertion and Wire Sizing for On-chip Interconnect Using Hierarchical Moment-Matching," In *Proceedings of IEEE/ACM Design Automation Conference*, pp. 502-506, 1999.

[86] C. J. Alpert, A. Devgan, and S. T. Quay, "Buffer Insertion for Noise and Delay Optimization," In *Proceedings of IEEE/ACM Design Automation Conference*, pp.362-367, 1998.

[87] A. Vittal, L. H. Chen, M. Marek-Sadowska, K.-P. Wang, and S. Yang, "Modeling Crosstalk in Resistive VLSI Interconnections," In *Proceedings of International Conference on VLSI Design*, pp.470-475, 1999.

[88] J. Cong, D. Z. Pan, and P. V. Srinivas, "Improved Crosstalk Modeling for Noise Constrained Interconnect Optimization," In *Proceedings of Asia and South Pacific Design Automation Conference*, pp.373-378, 2001.

[89] A. Vittal and M. Marek-Sadowska, "Crosstalk Reduction for VLSI," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 16, No. 3, pp.290-298, March 1997.

[90] T. Xiao and M. Marek-Sadowska, "Crosstalk Reduction by Transistor Sizing," In *Proceedings of Asia and South Pacific Design Automation Conference*, pp.137-140, 1999.

[91] H. Kawaguchi and T. Sakurai, "Delay and Noise Formulas for Capacitively Coupled Distributed RC Lines," In *Proceedings of Asia and South Pacific Design Automation Conference*, pp.35-43, 1998.

[92] A. Rubio, N. Itazaki, X. Zu, and K. Kinoshita, "An Approach to the Analysis and Detection of Crosstalk Faults in Digital VLSI Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 13, No. 3, pp.387-394, March 1994.

[93] A. Devgan, "Efficient Coupled Noise Estimation for On-Chip Interconnects," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp.147-151, 1997.

[94] A. Vittal, L. Chen, M. Marek-Sadowska, K.-P. Wang, and S. Yang, "Crosstalk in VLSI Interconnections," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 18, pp.1817-1824, 1999.

[95] B. N. Sheehan, "Predicting Coupled Noise in RC Circuits By Matching 1, 2, and 3 Moments," In *Proceedings of IEEE/ACM Design Automation Conference*, pp.532-535, 2000.

[96] M. Takahashi, M. Hashimoto, and H. Onodera, "An Analytic Crosstalk Noise Model Considering Coupling Location — Derivation and Evaluation —," In *Proceedings of IEICE General Conference*, 2001, to appear (In Japanese).

[97] H. B. Bakoglu, "Circuits, Interconnections, and Packaging for VLSI," Addison-Wesley Publishing Company, 1990.

[98] C. V. Kashyap, C. J. Alpert, and A. Devgan, "An Effective Capacitance Based Delay Metric for RC Interconnect," In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp.229-234, 2000.

[99] Arcadia Reference Manual. Synopsys, Inc., CA, 1999.

Publication List

Major Publications

1. M. Hashimoto and H. Onodera, "A Performance Optimization Method by Gate Resizing Based on Statistical Static Timing Analysis," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E83-A, No. 12, pp. 2558-2568, December 2000.

2. M. Hashimoto, H. Onodera and K. Tamaru, "A Power Optimization Method Considering Glitch Reduction by Gate Sizing," *IPSJ Transactions* , Vol. 40, No. 4, pp.1707-1716, April 1999(In Japanese).

3. M. Hashimoto, H. Onodera and K. Tamaru, "A Power and Delay Optimization Method using Input Reordering in Cell-Based CMOS Circuits," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E82-A No. 1, pp. 159-166, January 1999.

4. M. Hashimoto and H. Onodera, "Increase in Delay Uncertainty by Performance Optimization," In *Proceedings of IEEE International Symposium on Circuits and Systems*, 2001, to appear.

5. M. Hashimoto and H. Onodera, "Post-Layout Transistor Sizing for Power Reduction in Cell-Based Design," In *Proceedings of Asia and South Pacific Design Automation Conference*, pp. 359-365, 2001.

6. M. Hashimoto and H. Onodera, "A Statistical Delay-Uncertainty Analysis of the Circuits Path-Balanced by Gate/Transistor Sizing," In *Proceedings of ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pp. 34-37, 2000.

7. T. Iwahashi, T. Shibayama, M. Hashimoto, K. Kobayashi and H. Onodera, "Vector Quantization Processor for Mobile Video Communication," In *Proceedings of IEEE International ASIC/SOC Conference*, pp.75-79, 2000.

8. M. Hashimoto and H. Onodera, "A Performance Optimization Method by Gate Sizing using Statistical Static Timing Analysis," In *Proceedings of ACM International Symposium on Physical Design* pp.111-116, 2000.

9. M. Hashimoto and H. Onodera, "A Performance Optimization Method by Gate Resizing Based on Statistical Static Timing Analysis," In *Proceedings of the Ninth Workshop on Synthesis and System Integration of Mixed Technologies*, pp.115-121, 2000.

10. M. Hashimoto, H. Onodera and K. Tamaru, "A Practical Gate Resizing Technique Considering Glitch Reduction for Low Power Design," In *Proceedings of the 36th IEEE/ACM Design Automation Conference*, pp.446-451, 1999.

11. M. Hashimoto, H. Onodera and K. Tamaru, "A Power Optimization Method Considering Glitch Reduction by Gate Sizing," In *Proceedings of IEEE/ACM International Symposium on Low Power Electronics and Design*, pp.221-226, 1998.

12. M. Hashimoto, H. Onodera and K. Tamaru, "Input Reordering for Power and Delay Optimization," In *Proceedings of IEEE International ASIC Conference and Exhibit*, pp. 194-198, 1997.

日本語による口頭発表: Oral Presentations in Japanese.

1. 橋本 昌宜, 高橋 正郎, 小野寺 秀俊, "隣接位置を考慮した解析的クロストークノイズモデル, — 実回路への適用 —," 2001 年電子情報通信学会総合大会講演論文集, 2001, 発表予定.

2. 高橋 正郎, 橋本 昌宜, 小野寺 秀俊, "隣接位置を考慮した解析的クロストークノイズモデル — 導出と評価 —," 2001 年電子情報通信学会総合大会講演論文集, 2001, 発表予定.

3. 橋本 昌宜, 小野寺 秀俊, "パスバランス回路における遅延不確かさの統計的解析," 電子情報通信学会 VLSI 設計技術研究会 (デザインガイア), VLD2000-72, 2000.

4. 橋本 昌宜, 小野寺 秀俊, "パスバランス回路における遅延不確かさの統計的解析," 2000 年電子情報通信学会基礎・境界ソサイエティ大会講演論文集, No. A-3-9, pp.76, 2000.

5. 橋本 昌宜, "オンデマンドライブラリを用いた最適 LSI 設計手法," VDEC LSI デザイナーフォーラム, 出版予定.

6. 橋本 昌宜, 小野寺 秀俊, "セルベース設計における連続的トランジスタ寸法最適化による消費電力削減手法" 情報処理学会 DA シンポジウム 2000, pp.185-190, 2000.

7. 橋本 昌宜, 小野寺 秀俊, "静的統計遅延解析に基づいたゲート寸法最適化による回路性能最適化手法," 第 13 回 回路とシステム (軽井沢) ワークショップ, pp.137-142, 2000.

8. 橋本 昌宜, 小野寺 秀俊, "静的統計遅延解析を用いた最悪遅延時間計算手法," 2000 年電子情報通信学会総合大会講演論文集, 論文 No. A-3-13, pp.81, 2000.

9. 橋本 昌宜, 橋本鉄太郎, 西川亮太, 福田大輔, 黒田慎介, 菅俊介, 神原弘之, 小野寺 秀俊, "オンデマンドライブラリを用いたシステム LSI 詳細設計手法," 電子情報通信学会 VLSI 設計技術研究会, VLD99-112(ICD99-269), 2000.

10. 橋本 昌宜, 橋本 鉄太郎, 西川 亮太, 福田 大輔, 黒田 慎介, 菅 俊介, 神原 弘之, 小野寺 秀俊, "オンデマンドライブラリを用いたシステム LSI 詳細設計手法," 第 3 回 システム LSI 琵琶湖ワークショップ, pp.279-281, 1999.

11. 橋本 昌宜, 小野寺 秀俊, "スタンダードセルライブラリの駆動能力種類の追加による消費電力削減効果の検討," 1999 年電子情報通信学会基礎・境界ソサイエティ大会講演論文集, 論文 No. A-3-9, pp.52, 1999.

12. 橋本 昌宜, 小野寺 秀俊, 田丸 啓吉, "グリッチの削減を考慮したゲート寸法最適化による消費電力削減手法—レイアウト設計への適用—," 1998 年電子情報通信学会基礎・境界ソサイエティ大会講演論文集, 論文 No. A-3-5, pp.42, 1998.

13. 橋本 昌宜, 小野寺 秀俊, 田丸 啓吉, "グリッチの削減を考慮したゲート寸法最適化による消費電力削減手法," 情報処理学会 DA シンポジウム'98 論文集, pp.269-274, 1998.

14. 橋本 昌宜, 小野寺 秀俊, 田丸 啓吉, "論理シミュレーションを用いた消費電力見積もりの高精度化手法," 1998 年電子情報通信学会総合大会講演論文集, 論文 No. A-3-5, pp.91, 1998.

15. 橋本 昌宜, 小野寺 秀俊, 田丸 啓吉, "入力端子接続最適化による遅延時間と消費電力の最適化手法," 1997 年電子情報通信学会基礎・境界ソサイエティ大会講演論文集, 論文 No. A-3-15, pp.67, 1997.

16. 橋本 昌宜, 小野寺 秀俊, 田丸 啓吉, "入力端子接続最適化による消費電力削減手法," 情報処理学会 DA シンポジウム'97 論文集, pp.99-104, 1997.