POWER AND PERFORMANCE OPTIMIZATION OF STATIC CMOS CIRCUITS

WITH PROCESS VARIATION

Except where reference is made to the work of others, the work described in this dissertation is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

_____

Yuanlin Lu

Certificate of Approval:

_____	_____
Fa Foster Dai Associate Professor Electrical & Computer Engineering	Vishwani D. Agrawal, Chair James J. Danaher Professor Electrical & Computer Engineering
_____	_____
Charles E. Stroud Professor Electrical & Computer Engineering	Joe F. Pittman Interim Dean Graduate School

POWER AND PERFORMANCE OPTIMIZATION OF STATIC CMOS CIRCUITS WITH PROCESS VARIATION

Yuanlin Lu

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Doctor of Philosophy

Auburn, Alabama
August 4, 2007

POWER AND PERFORMANCE OPTIMIZATION OF STATIC CMOS CIRCUITS

WITH PROCESS VARIATION

Yuanlin Lu

_____

Signature of Author

_____

Date of Graduation

VITA

Yuanlin Lu, daughter of Rongchang Lu and Afeng Kong, was born in Nanjing, P. R. China. She attended Southeast University in 1995 and graduated with a Bachelor of Engineering degree in Electronic Information Engineering in 1999. She entered the Graduate School at Southeast University in 1999 and received the Master of Science degree in Circuit and System in 2002. In January 2004, she joined the Ph.D. program of the Department of Electrical and Computer Engineering, Auburn University.

DISSERTATION ABSTRACT

POWER AND PERFORMANCE OPTIMIZATION OF STATIC CMOS CIRCUITS WITH PROCESS VARIATION

Yuanlin Lu

Doctor of Philosophy, August 4, 2007
(M.S., Southeast University, 2002)
(B.S., Southeast University, 1999)

142 Typed Pages

Directed by Vishwani D. Agrawal

With the continuing trend of technology scaling, leakage power has become a main contributor to power consumption. Dual threshold (dual-V_{th}) assignment has emerged as an efficient technique for decreasing leakage power. In this work, a mixed integer linear programming (MILP) technique simultaneously minimizes the leakage and glitch power consumption of a static CMOS (Complementary Metal Oxide Semiconductor) circuit for any specified input-to-output critical path delay. Using dual-threshold devices, the number of high-threshold devices is maximized and a minimum number of delay elements is inserted to reduce the differential path delays below the inertial delays of the incident gates. The key features of the method are that the constraint set size for the MILP model is linear in the circuit size and a power-performance tradeoff is allowed.

Experimental results show 96%, 28% and 64% reductions of leakage power, dynamic power and total power, respectively, for the benchmark circuit C7552 implemented in BPTM 70nm CMOS technology.

Due to the exponential relation between subthreshold current and process parameters, such as the effective gate length, oxide thickness and doping concentration, process variations can severely affect both power and timing yields of the designs obtained by the MILP formulation. We propose a statistical mixed integer linear programming method for dual-V_{th} design that minimizes the leakage power and circuit delay in a statistical sense such that the impact of process variation on the respective yields is minimized. Experimental results show that 30% more leakage power reduction can be achieved by using a statistical approach when compared with the deterministic approach that has to consider the worst case in the presence of process variations.

Compared to subthreshold leakage, dynamic power is less sensitive to the process variation due to its linear dependency on the process parameters. However, the deterministic techniques using path balancing to eliminate glitches, becomes ineffective when process variation is considered. This is because the perfect hazard filtering conditions can easily be destroyed even by a small variation in some process parameters. We present a statistical MILP formulation to achieve a process-variation-resistant glitch-free circuit. Experimental results on an example circuit prove the effectiveness of this method.

ACKNOWLEDGMENTS

I would like to express my appreciation and sincere thanks to my advisor, Dr. Vishwani D. Agrawal, who guided and encouraged me throughout my studies. His advice and research attitude have provided me with a model for my entire future career. I also wish to thank my advisory committee members, Dr. Fa Foster Dai and Dr. Charles E. Stroud for their guidance and advice on this work.

Appreciation is expressed to Badhri Uppiliappan who gave me a great help during my internship in Analog Device Inc.

I also appreciate those who have made contributions to my research. Thanks to Jins Alexander, Hillary Grimes, Kyungseok Kim, Khushboobenumesh Sheth, Fan Wang and Nitin Yogi for their cooperation and helpful discussions throughout the course of this research.

Finally, I would like to thank, although this is too weak a word, my parents and sister, all the other family members and my friends for their continual encouragement and support throughout this work.

Style manual or journal used: Bibliography follows those of the transactions of the Institute of Electrical and Electronics Engineers and is sorted in alphabetical order.

Computer software used: Microsoft Word 2003.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

The primary contribution of this work is a new design methodology to minimize the total power consumption in a static CMOS (Complementary Metal Oxide Semiconductor) circuit. A mixed integer linear programming (MILP) formulation is proposed to optimize leakage power and dynamic glitch power, without reducing circuit performance, by dual-V_{th} assignment, path balancing and gate sizing. To consider the process variation, statistical delay and leakage models are adopted to optimize power consumption in a statistical sense such that the impact of process variation on the power and timing yields is minimized.

1.1 Motivation

With the continuous increase of the density and performance of integrated circuits due to the scaling down of the CMOS technology, reducing power dissipation becomes a serious problem that every circuit designer has to face.

1.1.1 Leakage Power

In the past, the dynamic power dominated the total power dissipation of a CMOS device. Since dynamic power is proportional to the square of the power supply voltage, lowering the voltage reduces the power dissipation. However, to maintain or increase the performance of a circuit, its threshold voltage should be decreased by the same factor,

which causes the subthreshold leakage current of transistors to increase exponentially and make it a major contributor to power consumption.

To reduce leakage power, many techniques have been proposed, including transistor sizing [45, 72], multi-V_{th} [12, 19, 103], dual-V_{th} [31, 45, 70, 72, 96-101], optimal standby input vector selection [69, 84], transistor stacking [64, 65, 106], body bias [10, 91], *etc*. As the threshold voltage (V_{th}) of transistors in a CMOS logic gate is increased, the leakage current is reduced but the gate slows down. Dual-V_{th} assignment is an efficient technique for leakage reduction. The basic idea is utilizing the timing slack on non-critical paths to minimize the leakage power by assigning high V_{th} to some or all gates on non-critical paths.

1.1.2 Glitch Power

Glitches as unnecessary signal transitions account for 20%-70% of the dynamic switching power [20]. To eliminate glitches, a designer can adopt techniques of hazard filtering [7, 38, 46, 83, 104] and path balancing [8, 46, 74]. In Hazard filtering, gate sizing or transistor sizing is used to increase the gate's inertial delay to filter out the glitches. An obvious disadvantage of such hazard filtering, when used alone, is that it may increase the circuit delay due to the increase of the gate delay. Alternatively, any given performance can be maintained by path delay balancing, although the area overhead and additional power consumption of the inserted delay elements can become a major concern. The best way to eliminate glitches is to combine these two techniques [8].

1.1.3 Process Variation

The increase in variability of several key process parameters can significantly affect the design and optimization of low power circuits in the nanometer regime [61]. Due to the exponential relation of leakage current with some process parameters, such as the effective gate length, oxide thickness and doping concentration, process variations can cause a significant increase in the leakage current. There are two principal components of leakage current. Gate leakage is most sensitive to the variation in oxide thickness (T_{ox}), while the subthreshold current is extremely sensitive to the variation in effective gate length (L_{eff}), oxide thickness (T_{ox}) and doping concentration (N_{dop}). Compared to gate leakage, subthreshold leakage is more sensitive to parameter variations [66].

Dynamic power is normally much less sensitive to the process variation because of its approximately linear dependency on the process parameters. However, any deterministic path balancing technique used for eliminating glitches becomes less effective under process variation, since the perfect hazard filtering conditions can be easily corrupted even with a small variation in some process parameters. To make the glitch-free circuits optimized by path balancing resistant to process variations, a statistical delay model is developed in this work.

1.2 Problem Statement

The problem solved in this work is: *Find a deterministic mixed integer linear programming (MILP) formulation to optimize the total power consumption by dual threshold voltage (dual-V_{th}) assignment, path balancing and gate sizing. Further, derive*

a statistical mixed integer linear programming formulation to minimize the impact of process variations on the optimal leakage and dynamic glitch power.

1.3 Original Contributions

In this dissertation, we first propose a deterministic mixed integer linear programming (MILP) formulation to minimize the leakage and dynamic power consumption of a static CMOS circuit for a given performance. In a dual-threshold circuit this method maximizes the number of high-threshold devices and simultaneously eliminates glitches by balancing paths with the smallest number of delay elements. Gate sizing is also considered to further minimize the dynamic switching power by reducing the loading capacitances of gates.

Since leakage exponentially depends on some key process parameters, it is very sensitive to process variations. We treat gate delay and leakage current as random variables to reflect the impact of process variation. A mixed integer linear programming (MILP) method for dual-V_{th} design is proposed to minimize the leakage power and circuit delay in a statistical sense such that the effect of process variation on the respective yields is minimized. Two types of yields are considered. Leakage yield refers to the probability of an optimized circuit retaining the leakage current below the specified value in the presence of random process variations. Similarly, timing yield is the probability of the critical path delay staying below the specification. The experimental results show that 30% more leakage power reduction can be achieved by using the statistical approach, referred to as statistical MILP, when compared with the deterministic approach.

Glitch-free circuits optimized by path balancing are also quite sensitive to process variations. We further extend the statistical MILP formulation to optimize the dynamic switching power considering process variation and achieve process-variation-resistant glitch-free circuits.

1.4 Organization of the Dissertation

In Chapter 2, the basic components of power consumption in a static CMOS circuit are first discussed, followed by a survey of the relevant published literature on low power design techniques at the gate level. Chapter 3 proposes an original mixed integer linear programming (MILP) method for total power minimization by dual-V_{th} assignment and path balancing. To consider process variation, statistical MILP optimization of leakage power and dynamic glitch power are presented in Chapter 4 and Chapter 5, respectively. In Chapter 6, experimental results are presented. Finally, a conclusion and recommendations for future work are given in Chapter 7.

CHAPTER 2 PRIOR WORK: TECHNIQUES FOR LOW POWER DESIGN

2.1 Components of Power Consumption

Power consumption in a static CMOS circuit basically comprises three components: dynamic switching power, short circuit power and static power. Compared to the other two components, short circuit power normally can be ignored in submicron technology.

2.1.1 Dynamic Power

Dynamic power is due to charging and discharging the loading capacitances. It can be expressed by the following equation [73]:

$$P_{dyn} = \frac{1}{2} C_L V_{dd}^{\ 2} \cdot A \cdot F \tag{2.1}$$

where

- C_L is the loading capacitances, including the gate capacitance of the driven gate, the diffusion capacitance of the driving gate and the wire capacitance;
- V_{dd} is the power supply voltage;
- A is the switching activity;
- F is the circuit operating frequency.

Equation (2.1) shows that dynamic switching power is directly proportional to the switching activity, A, or the number of signal transitions. More the signal transitions,

higher is the dynamic power consumption. After a transition is applied at the input, the output of a gate may have multiple transitions before reaching a steady state (see Figure 2.9(a)). Among these transitions, at most one is the essential transition, and all others are unnecessary transitions that are called *glitches or hazards.* Hence, dynamic power is composed of two parts, logic switching power which is contributed by the necessary signal transitions for logic functions, and glitch power which is caused by glitches or hazards.

2.1.2 Leakage Power

The leakage current of a transistor is mainly the result of reverse-biased PN junction leakage, subthreshold leakage and gate leakage as illustrated in Figure 2.1.

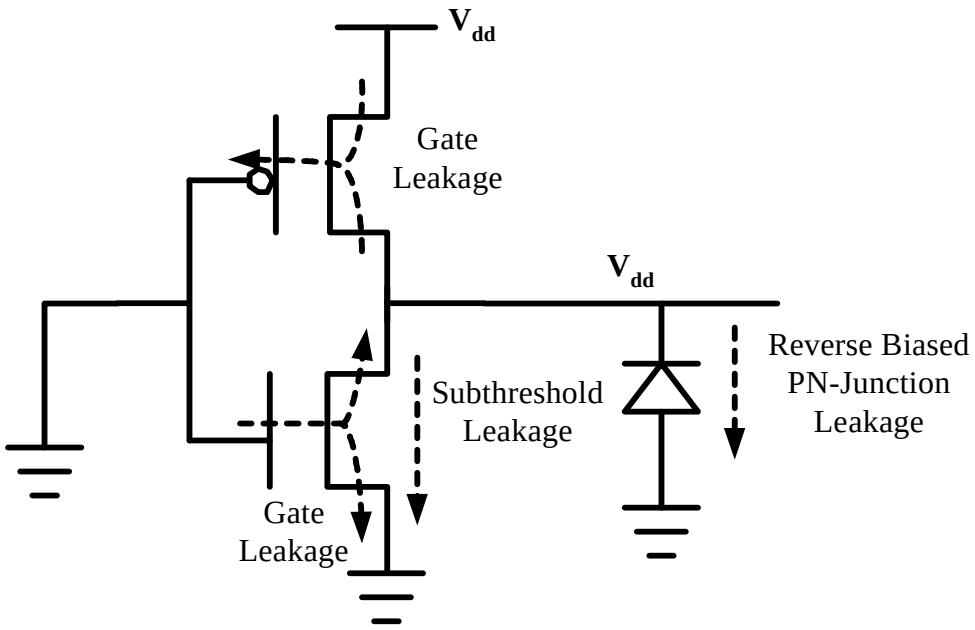


Figure 2.1 Leakage currents in an inverter.

In submicron technology, the reverse-biased PN junction leakage is much smaller than subthreshold and gate leakage and hence can be ignored. The subthreshold leakage is the weak inversion current between source and drain of an MOS transistor when the gate voltage is less than the threshold voltage [99]. It is given by [42]:

$$I_{sub} = \mu_0 C_{ox} \frac{W}{L_{eff}} V_T{}^2 e^{1.8} \exp\left(\frac{V_{gs} - V_{th}}{nV_T}\right) \cdot \left(1 - \exp\left(\frac{-V_{ds}}{V_T}\right)\right) \tag{2.2}$$

where μ_0 is the zero bias electron mobility, C_{ox} is the oxide capacitance per unit area, n is the subthreshold slope coefficient, V_{gs} and V_{ds} are the gate-to-source voltage and drain-to-source voltage, respectively, V_T is the thermal voltage, V_{th} is the threshold voltage, W is the channel width and L_{eff} is the effective channel length, respectively. Due to the exponential relation between I_{sub} and V_{th}, an increase in V_{th} sharply reduces the subthreshold current.

Gate leakage is the oxide tunneling current due to the low oxide thickness and the high electric field which increases the possibility that carriers tunnel through the gate oxide. Tunneling current will become a factor and may even be comparable to subthreshold leakage when oxide thickness is less than 15-20Å [102]. Unlike subthreshold leakage, which only exists in weakly turned-off transistors, gate leakage always exists no matter whether the transistor is turned on or turned off [100]. Equation (2.3) gives the expression of the gate leakage [64].

$$I_{gate} = W_{eff} L_{eff} A \left(\frac{V_{ox}}{T_{ox}}\right)^2 \exp\left(\frac{-B\left(1 - \left(1 - \frac{V_{ox}}{\phi_{ox}}\right)^{\frac{3}{2}}\right)}{\frac{V_{ox}}{\phi_{ox}}}\right) \tag{2.3}$$

where V_{ox} is the potential drop across the thin oxide, Φ_{ox} is the barrier height for the tunneling particle (electron or hole), and T_{ox} is the oxide thickness. A and B are physical parameters given by [64],

$$A = \frac{q^3}{16\pi^2 h\phi_{ox}} \text{ and } B = \frac{4\sqrt{2m}\phi_{ox}^{\frac{3}{2}}}{3hq},$$

where m is the effective mass of the tunneling particle, q is the electronic charge, and h is the reduced Plank's constant. The oxide thickness T_{ox} decreases with the technology scaling to avoid the short channel effects. Equation (2.3) shows that gate leakage increases significantly with the decrease of T_{ox}.

In this work, we use BPTM (Berkeley Predictive Technology Models) 70nm technology [1] to implement our designs. Since BPTM 70nm technology is characterized by BSIM3.5.2, which cannot correctly model gate leakage, gate leakage is omitted in this work, and all the techniques discussed in Section 2.2 aim at subthreshold leakage reduction.

2.2 Techniques for Leakage Reduction

Leakage is becoming comparable to dynamic switching power with the continuous scaling down of CMOS technology. To reduce leakage power, many techniques have been proposed, including dual-V_{th}, multi-V_{th}, optimal standby input vector selection, transistor stacking, and body bias.

2.2.1 Dual-V_{th} Assignment

Dual-V_{th} assignment is an efficient technique for leakage reduction. In this method, each cell in the standard cell library has two versions, low V_{th} and high V_{th}. Gates with low V_{th} are fast but have high subthreshold leakage, whereas gates with high V_{th} are slower but have much reduced subthreshold leakage. Traditional deterministic approaches for dual-threshold assignment utilize the timing slack of non-critical paths to assign high V_{th} to some or all gates on those non-critical paths to minimize the leakage power.

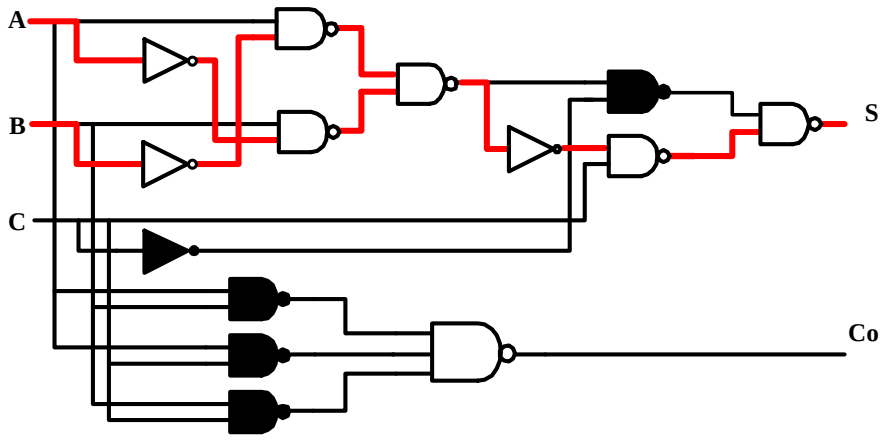


Figure 2.2 An example dual-V_{th} circuit.

Figure 2.2 gives an example dual-V_{th} circuit. The bold lines represent the critical paths. To keep the highest circuit performance, all gates on the critical paths are assigned low V_{th} (white gates), while some gates on those non-critical paths can be assigned high V_{th} (black gates) to reduce the leakage since there are timing slacks left on those non-critical paths. Based on the techniques used for determining which gates on non-critical paths should be assigned high $V_{th,}$ the dual-V_{th} approaches can be basically divided into

two groups: heuristic algorithms [45, 72, 96-101] and linear programming algorithms [31, 70]. Among heuristic algorithms, the *backtracking* algorithm [97, 98] used to determine the dual-V_{th} assignment only gives a possible solution, not usually an optimal one (see example in Figure 3.8 in Section 3.4). Because the backtracking search direction for non-critical paths is always from primary outputs to primary inputs, the gates close to the primary outputs have a higher priority for high V_{th} assignment, even though their leakage power savings may be smaller than those of gates close to the primary inputs. In [96], dual-V_{th} assignment is described as a constrained 0-1 programming problem with non-linear constraint functions. Wang *et al*. use a heuristic algorithm based on circuit graph enumeration to solve this problem. Although their swapping algorithm tries to avoid the local optimization, a global optimization still can not be guaranteed. Unlike a heuristic algorithm that can only guarantee a locally optimal solution, a linear programming (LP) formulation ensures a global optimization by describing both the objective function and constraints as linear functions. Nguyen *et al.* [70] use LP to minimize the leakage and dynamic power by gate sizing and dual-V_{th} device assignment. The optimization work is separated into several steps. An LP is first used to distribute slack to gates with the objective of maximizing total power reduction. Then, an independent algorithm is needed to resize gates and assign threshold levels. This means that in [70] LP still needs the assistance of a heuristic algorithm to complete the optimization. The method of [31] also uses MILP to optimize the total power consumption by dual-threshold assignment and gate sizing.

Dual-V_{th} assignment can reduce leakage in both active and standby modes since some gates remain idle even when the whole circuit or system is in the active mode. But

the effectiveness of this method depends on the circuit structure. A symmetric circuit with many critical paths leaves a much reduced optimization space for leakage reduction.

2.2.2 Multi-Threshold-Voltage CMOS

A Multi-Threshold-Voltage CMOS (MTCMOS) circuit [12, 19, 103] is implemented by inserting high V_{th} transistors between the power supply voltage and the original transistors of the circuit [68]. Figure 2.3(a) shows a schematic of a MTCMOS NAND gate. The original transistors are assigned low V_{th} to enhance the performance while high-V_{th} transistors are used as sleep controllers. In active mode, SL is set low and sleep control high-V_{th} transistors (MP and MN) are turned on. Their on-resistance is so small that VSSV and VDDV can be treated as almost being equal to the real power supply. In the standby mode, SL is set high, MN and MP are turned off and the leakage current is low. The large leakage current in the low-V_{th} transistors is suppressed by the small leakage in the high-V_{th} transistors. By utilizing the sleep control high-V_{th} transistors, the requirements for high performance in active mode and low static power consumption in standby mode can both be satisfied.

To reduce the area, power and speed overhead contributed by the sleep control high-V_{th} transistors, only one high-V_{th} transistor is needed. Figure 2.3(b) and 2.3(c) show the PMOS insertion MTCMOS and NMOS insertion MTCMOS. NMOS insertion MTCMOS is preferred because for any given size, an NMOS transistor has smaller on-resistance than a PMOS transistor [100].

Compared to the dual-V_{th} technique, MTMOS can only reduce leakage in the standby mode and has additional area-, power-, and speed overheads.

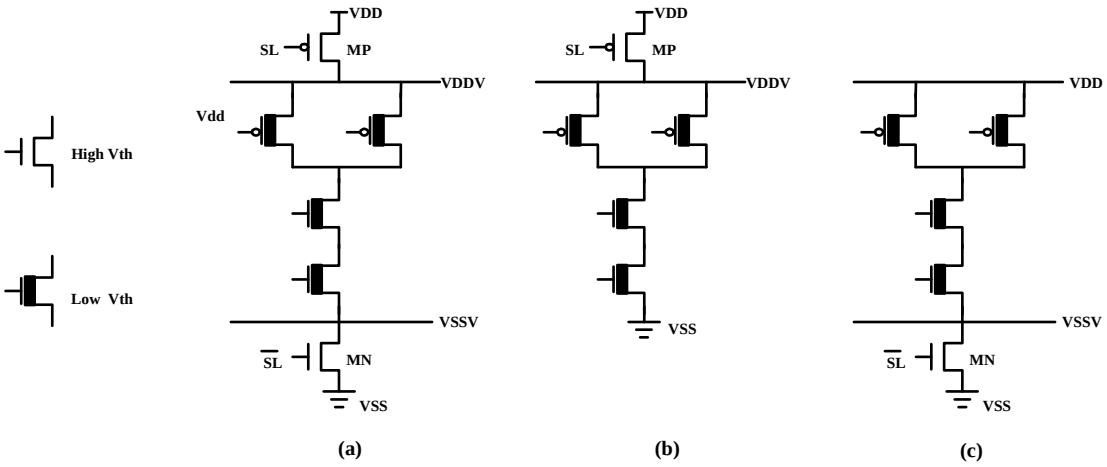


Figure 2.3 Schematic of MTCMOS, (a) original MTCMOS, (b) PMOS insertion MTCMOS, (c) NMOS insertion MTCMOS.

2.2.3 Adaptive Body Bias

The threshold voltage of a short-channel NMOSFET can be expressed by the following equation [47].

$$V_{th} = V_{th0} + \gamma\left(\sqrt{\phi_s - V_{bs}} - \sqrt{\phi_s}\right) - \theta_{DIBL} V_{dd} + \Delta V_{NW} \tag{2.4}$$

where V_{th0} is the threshold voltage with a zero body bias, Φ_S, γ and θ_{DIBL} are constants for a given technology, V_{bs} is the voltage applied between the body and source of the transistor, ΔV_{NW} is a constant that models narrow width effect, and V_{dd} is the supply voltage. Equation (2.4) shows that a reverse body bias leads to an increase of the threshold voltage and a forward body bias decreases the threshold voltage.

Leakage power reduction can be achieved by dynamically adjusting the threshold voltage through adaptive body bias according to the different operation modes. In the active mode, forward body (or zero) bias is used to reduce the threshold voltage, which results in a higher performance. In the standby mode, leakage power is greatly reduced by

the optimal reverse body bias, which increases threshold voltages. The basic scheme of an adaptive-body-biased inverter is shown in Figure 2.4 [100].

Similar to the MTCMOS, adaptive body bias [11, 13, 28, 54, 63, 90] only reduces the leakage power in the standby mode. With the continuous technology scaling, the optimal reverse body bias becomes closer to the zero body bias and thus the technique of adaptive body bias becomes less effective [44].

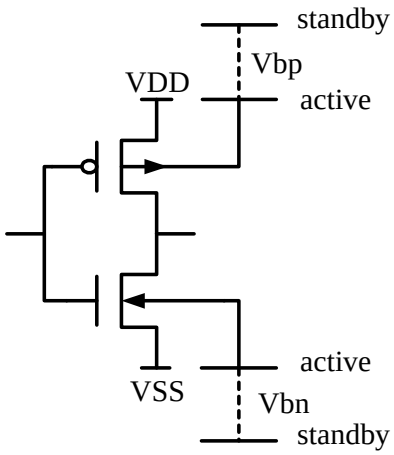


Figure 2.4 Scheme of an adaptive body biased inverter.

2.2.4 Transistor Stacking

The two serially-connected devices in the off state have significantly lower leakage current than a single off device. This is called the stacking effect [64, 65, 106]. In Figure 2.5(b), when both M1 and M2 are turned off, V_m has a positive value due to the leakage current flowing through M1 and M2. Assuming the bodies of M1 and M2 are both connected to the ground, V_{bs} of M1 becomes negative and leads to an increase of M1's threshold voltage. At the same time, V_{gs} and V_{ds} of M1 are both reduced. According to equation (2.2), the subthreshold leakage in M1 is decreased sharply and suppresses the

relative larger leakage current in M2. On the contrary, V_m in Figure 2.4(a) is always equal to zero and has no effect on V_{bs}, V_{gs} and V_{ds} of M and hence on its subthreshold leakage.

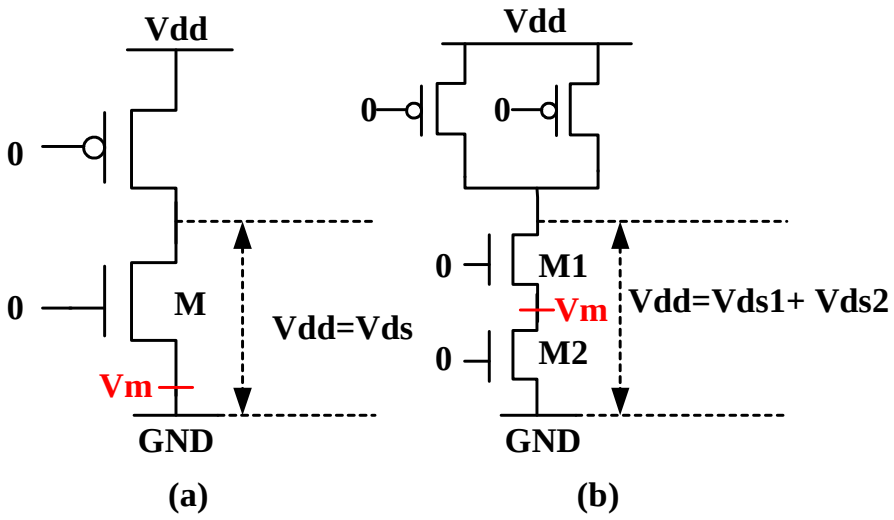


Figure 2.5 Comparison of leakage for (a) one single off transistor in an inverter and (b) two serially-connected off transistors in a 2-input NAND gate.

With transistor stacking [40, 51, 55], by replacing one single off transistor with a stack of serially-connected off transistors, leakage can be significantly reduced. The disadvantages of this technique are also obvious. Such a stack of transistors causes either performance degradation or more dynamic power consumption.

2.2.5 Optimal Standby Input Vectors

Subthreshold leakage current depends on the vectors applied to the gate inputs because different vectors cause different transistors to be turned off. From the illustration in Section 2.2.4, a 2-input NAND gate has the smallest subthreshold leakage due to the stacking effect when the input vector is ‘00’. When a circuit is in the standby mode, one

could carefully choose an input vector and let the total leakage in the whole circuit to be minimized [6, 22, 32, 52, 69, 84]. Gao *et al.* in [32] model leakage current by means of linearized pseudo-Boolean functions. An exact ILP model was first discussed to minimize leakage with respect to a circuit's input vector. A fast heuristic MILP was then proposed to selectively relax some binary constraints of the ILP model to make a tradeoff between runtime and optimality.

2.2.6 Power cutoff

Yu and Bushnell [108, 109] present a novel active leakage power reduction method called the *dynamic power cutoff technique* (DPCT). The power supply to each gate is only connected in its switching window, during which the gate makes its transition within a clock cycle. The circuit is optimally partitioned into groups based on the *minimal switching window* (MSW) of gates and power cutoff transistors are inserted into each group to control the power connection of that group. Since the power supply of each gate is only turned on during a small timing window within a clock cycle, significant active leakage reduction can be achieved. One key of this leakage reduction technique is the implementation of the cutoff transistors, which can be either implemented by high-V_{th} transistors as discussed in Section 2.2.2, or by low-V_{th} transistors that are overdriven by a power supply larger than V_{dd} for PMOS cutoff transistors or lower than V_{ss} for NMOS cutoff transistors.

2.3 Techniques for Dynamic Power Reduction

Dynamic power is comprised of logic switching power and glitch power, and can be expressed by the following equation [73].

$$P_{dyn} = \frac{1}{2} C_L V_{dd}^{\ 2} \cdot A \cdot F \tag{2.4}$$

To reduce dynamic power at a specified operating frequency F, we can either reduce the dynamic power consumption per logic transition which is determined by loading capacitances C_L, and power supply V_{dd}, or reduce the number of logic transitions in the circuit represented by switching activity A.

2.3.1 Logic Switching Power Reduction

2.3.1.1 Dual power supply

Reducing the supply voltage, or voltage scaling [15, 23, 27, 29, 107], is the most effective technique for dynamic power reduction because dynamic power is proportional to the square of the power supply. Similar to the dual-V_{th} approach, the dual V_{dd} technique assigns high V_{dd} to all the gates on the critical paths and low V_{dd} to some of the gates on the non-critical paths. When a gate operating at a lower V_{dd} directly drives a higher V_{dd} gate, a level converter is required to avoid the undesirable short circuit power in that higher V_{dd} gate due to the possible large DC current caused by the low voltage fanin. Since the level converters contribute additional power, minimizing the number of level converters is also important in voltage scaling [9].

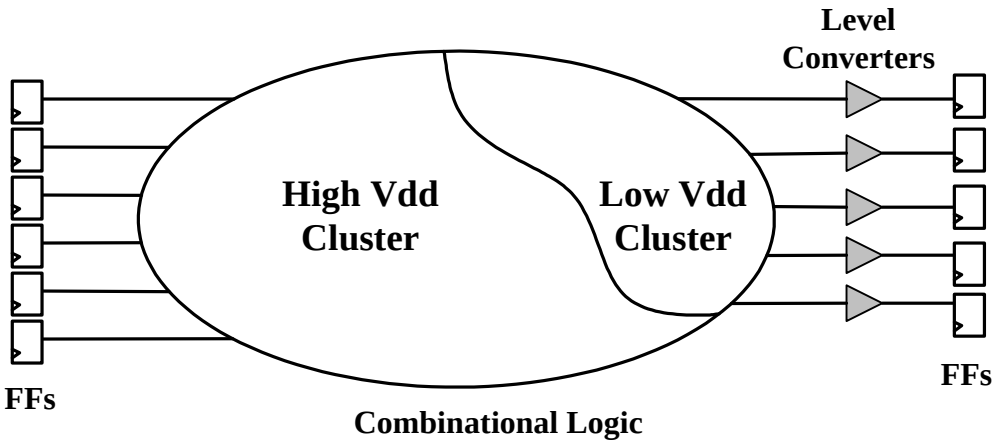


Figure 2.6 Scheme of cluster voltage scaling.

Clustered voltage scaling (CVS) [94] is an effective voltage scaling technique. The basic idea is shown in Figure 2.6 [9]. The instances of low V_{dd} gates driving high V_{dd} gates are not allowed and level converters are only used to convert low voltage signals to high voltage as inputs to flip-flops (FFs) such that the total number of level converters is minimized.

In contrast to CVS, extended clustered voltage scaling (ECVS) [95] allows level conversion anywhere and the supply voltage assignment to the gates is much more flexible. Thus greater dynamic power saving can be achieved compared to the CVS. The algorithm of ECVS is more complicated than that of CVS, since CVS may use a backtracking algorithm to determine just two clusters: one high V_{dd} cluster and the other a low V_{dd} cluster. Figure 2.7 gives an example circuit whose dynamic power is optimized by ECVS. The bold lines represent the critical paths.

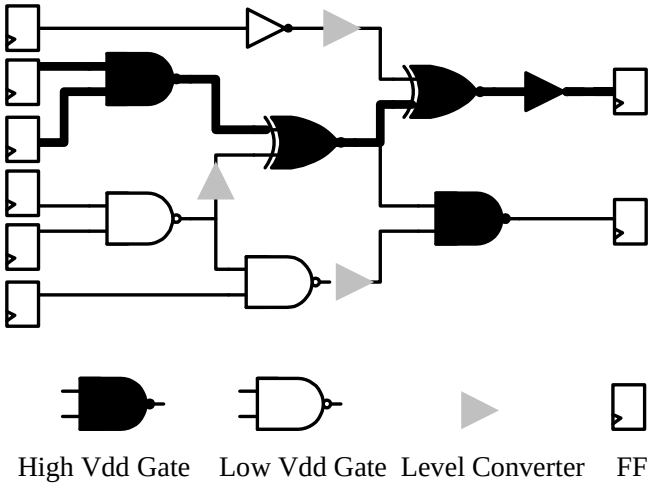


Figure 2.7 Example circuit for illustrating ECVS.

2.3.1.2 Gate sizing

Non-critical paths have timing slack and the delays of some gates on these paths can be increased without affecting the performance. Since the lengths of devices (transistors) in a gate are usually minimal for a high speed application, the gate delay can be increased by reducing the device width. As a result, the dynamic power is accordingly decreased due to smaller loading capacitance C_L, which is proportional to the device size.

Gate sizing is a technique that determines device widths for gates. Traditional gate sizing approaches use Elmore delay models in a polynomial formulation. Heuristics-based greedy approaches [23-25, 67, 78, 86, 101] can be used to solve such a polynomial problem. In general, a heuristic algorithm is relatively fast but cannot guarantee a global optimal.

The gate delay with respect to its device size, used in [23-25, 67, 78, 101], is generally given by the following equation,

$$d_i = gd_i + C_i \frac{C_{out_i}}{GS_i} \tag{2.5}$$

where, d_i is the delay of the gate, gd_i is the intrinsic gate delay of gate i, C_i is a constant, C_{outi} is the fanout load of gate i and GS_i is the width of the gate i. The total loading capacitance C_{outi} is determined based on the fanout of the gate and is given as [78],

$$C_{out_i} = \sum_{j \in FO(i)} (C_{wire_{ij}} + C \cdot GS_j) \tag{2.6}$$

where, $FO(i)$ is the set of gates that form the fan-outs for gate i, C_{wireij} is the capacitance of the wire connecting gates i and j and C is a constant. When ignoring the wiring capacitance, Equation (2.5) can be rewritten as (2.7).

$$d_i = gd_i + k_i \sum_{j \in FO(i)} \frac{GS_j}{GSi} \tag{2.7}$$

where $k_i = C \cdot C_i$.

A linear programming method is proposed [14] in which a piecewise linear delay model is adopted to achieve a global optimal solution. A non-linear programming approach [59] gives the most accurate optimal solution but at a cost of long run times.

2.3.1.3 Transistor sizing

The basic idea of transistor sizing is exactly the same as that of gate sizing except that in gate sizing all the transistors in one gate are sized together with the same factor but in transistor sizing each transistor can be sized independently.

Gate intrinsic delay actually depends on the current and previous input vectors which determine the internal IO path (from the gate inputs to gate output). Different internal IO paths have different on-resistances that cause distinct path delays (gate intrinsic delays).

For a gate on a critical path, only part of its transistors contribute the largest intrinsic gate delay, so the remaining transistors still can be sized to reduce the capacitances. In gate sizing, gd_i, the intrinsic gate delay of gate i in Equation (2.5) and (2.7) is a fixed value which makes it impossible to differentiate among the internal IO paths. On the contrary, transistor sizing [16, 43, 85, 105] explores the maximum possible optimization space by sizing transistors independently.

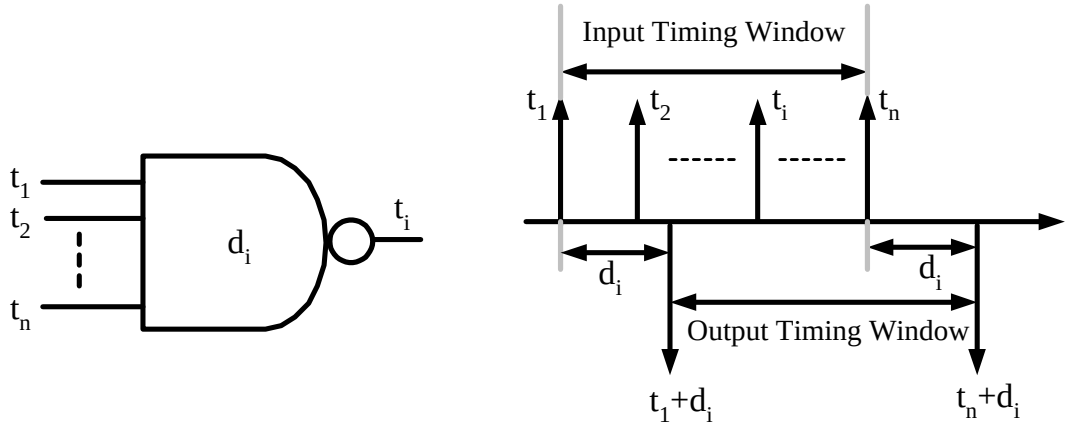
2.3.2 Glitch Power Elimination

When transitions are applied at inputs of a gate, the output may have multiple transitions before reaching a steady state (Figure 2.9(a)). Among these, at most one is the essential transition, and all others are unnecessary transitions often called *glitches* or *hazards*. Because switching power consumed by the gate is directly proportional to the number of output transitions, glitches reportedly account for 20%-70% dynamic power [20].

Agrawal *et al.* [8] prove that a combinational circuit is minimum transient energy design, i.e., there is no glitch at the output of any gate, if the difference of the signal arrival times at every gate's inputs remains smaller than the inertial delay of the gate, which is the time interval that elapses after a primary input change before the gate can produce a change at its output. This condition is expressed by the following inequality:

$$t_n - t_1 < d_i \tag{2.8}$$

where we assume t_1 is the earliest arrival time at inputs, t_n is the most delayed arrival time at another input, and d_i is gate's inertial delay, as shown in Figure 2.8. The interval $t_n - t_1$ is referred to as the gate input/output timing window [74].



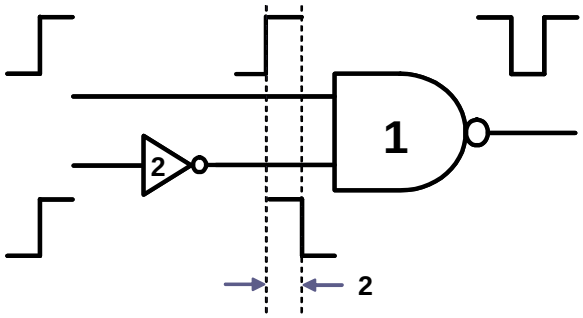
(a) a n-input NAND gate (b) timing window for the inputs and output of gate in (a)

Figure 2.8 Timing window for an n-input NAND gate.

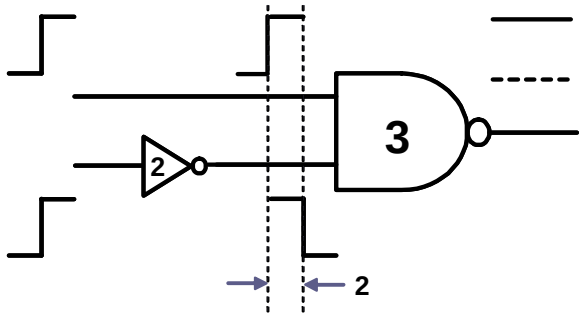
To satisfy inequality (2.8), we can either increase the inertial delay d_i (*hazard filtering* or *gate/transistor sizing*) or decease the path delay difference $t_n - t_1$ (*path balancing*). Figures 2.9(b) and 2.9(c) illustrate these procedures for the gate of Figure 2.9(a).

2.3.2.1 Hazard filtering

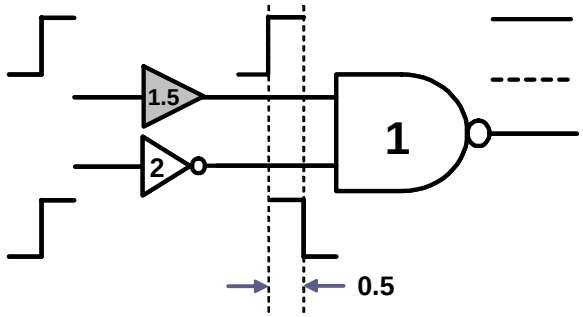
In hazard filtering, the inertial gate delay is increased to be larger than the timing window by gate/transistor sizing [7, 33, 104], so that the gate itself acts as a hazard filter. Figure 2.9(a) shows that the timing window is 2 units, which is larger than the inertial gate delay of 1 unit. Glitches are generated at this gate's output. In Figure 2.9(b), the inertial gate delay is increased from 1 unit to 3 units for hazard filtering and glitches are removed.



(a) Glitch at the output of one NAND gate



(b) Glitch elimination by hazard filtering



(c) Glitch elimination by path delay balancing

Figure 2.9 Glitch elimination methods, (a) glitches at the output of a NAND gate, (b) glitch elimination by hazard filtering, and (c) glitch elimination by path delay balancing.

In [7], hazard filtering is applied to a full adder circuit and 42% dynamic power is reduced. The glitch-free circuit has gates whose speed is decreased to 20% of their original value but with little reduction in overall speed of the circuit. This is because those gates are mainly on non-critical paths and do not contribute much to the critical path delay of the circuit.

2.3.2.2 Path balancing

In path balancing, the timing window, $t_n - t_1$, is reduced to be less than the inertial gate delay by inserting delay elements [8, 46, 74] on the faster input paths. In Figure 2.9(c), a 1.5 unit delay is inserted on the faster input path and reduces the timing window to be 0.5 units, which is less than the inertial delay of the gate. Hence glitches are eliminated at the gate output. Since delay elements contribute additional power, the low-power delay elements should be selected. Section 3.3 gives a detailed discussion of two popular delay elements. In [92], the authors use resistive-feed-through cells to implement delays. This technique can eliminate glitch power but at a cost of huge area overhead which is contributed by the large inserted resistance. Raja, *et al.* [75, 76] propose a path balancing technique based on a new variable-input-delay logic or a new design style where logic gates have different delays along I/O paths through them. Therefore, a glitch free circuit can be designed without inserting delay elements. But, the design of this type of gates has technology limitations due to the amount of differential delay that can be realized.

Hazard filtering or gate/transistor sizing, when used alone, can increase the overall input-to-output delay since some gates on critical paths have to increase their inertial

delays to eliminate glitches. On the other hand, due to the upper bound of the gate delays in a specific technology, gate delay cannot be increased without bound, so some of glitches cannot be removed in a circuit that has a large logic depth or large critical path delay. Hazard filtering or gate/transistor sizing usually cannot guarantee 100% glitch elimination. Path balancing does not increase the delay, and guarantees to eliminate all glitches, but requires insertion of delay elements that contribute power and area overheads. A combination of the two procedures [8, 46] can give an optimum design.

2.3.2.3 Hazard-free circuit design

In an asynchronous system, some control signals should be very clean and without any hazard (glitch). Hazard-free circuits can be adopted to generate such signals. The multiplexer circuit in Figure 2.10(a) has a glitch at its output when A changes from 1 to 0 while both B and C are 1. By adding a redundant gate (gray shaded gate) in Figure 2.10(b), this hazard can be eliminated [18].

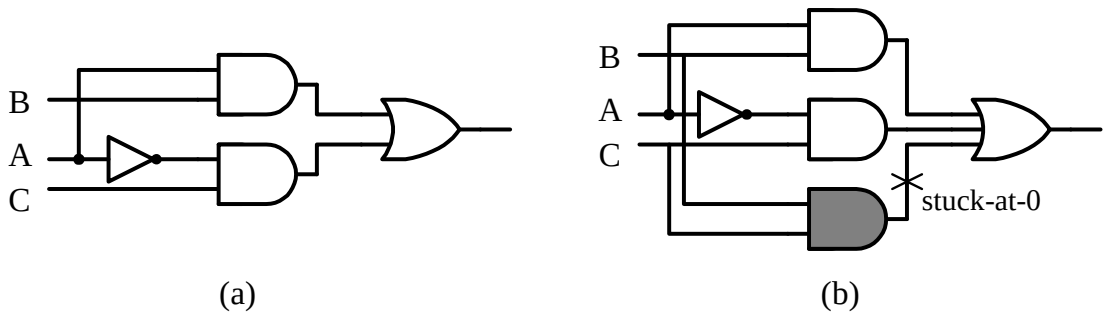


Figure 2.10 Using redundant implicant to eliminate hazards, (a) a multiplexer with hazards, and (b) a redundant implementation of multiplier free from certain hazards.

Besides the area and power overhead, an additional disadvantage of this method is that it introduces redundant stuck-at faults, such as the one shown in Figure 2.10(b). This

fault cannot be tested. On the other hand, if this fault is present then the circuit loses the hazard-suppression capability. Another disadvantage is that such method cannot guarantee to eliminate all the hazards caused by multiple input-signal transitions.

In [71], authors present a new method for two-level hazard-free logic minimization of Boolean functions with multiple-input changes. Given an incompletely-specified Boolean function, this method produces a minimal sum-of-products implementation, which is hazard-free for a given set of multiple-input changes, if such a solution exists. Overhead due to hazard-elimination is shown to be negligible.

2.4 Power Optimization with Process Variation

2.4.1 Leakage Minimization with Process Variation

Due to the exponential dependency of subthreshold leakage on some key process parameters, the increased presence of process parameter variations in modern designs has accentuated the need to consider the impact of statistical leakage current variations during the design process. Up to three times change in the amount of subthreshold leakage current is observed with ±10% variation in the effective channel lengths of transistors [66, 79].

Statistical analysis and estimation of leakage power considering process variation are presented in [21, 80, 81]. A lognormal distribution is used to approximate the leakage current of each gate and the total chip leakage is determined by summing up the lognormals [21].

Variation of process parameters not only affects the leakage current but also changes the gate delay, degrading either one or both, power and timing yields of an optimized

design. To minimize the effect of process variation, some techniques [26, 61, 89] statistically optimize the leakage power and circuit performance by dual-V_{th} assignment. Leakage current and delay are treated as random variables. A dynamic programming approach for leakage optimization by dual-V_{th} assignment has been proposed [26], which uses two pruning criteria that stochastically identify pareto-optimal solutions and prune the sub-optimal ones. Another approach [61] solves the statistical leakage minimization problem using a theoretically rigorous formulation for dual-V_{th} assignment and gate sizing. Liu *et al.* [53] reduce leakage power by dual-V_{th} design in a probabilistic analysis method. They assume a lower V_{th}, predetermined by the timing requirements, and an optimal higher V_{th} is then selected in the presence of variability. The probabilistic model demonstrates that the true average leakage power is three times as large as that predicted by a non-probabilistic model.

2.4.2 Glitch Power Optimization with Process Variation

The delay of the gate is modeled as a fixed value in the deterministic methods discussed in Section 2.3.2. In reality, however, process variations make the delays to be random variables, generally assumed to have Gaussian distributions. The glitch filtering condition of inequality (2.8) cannot be guaranteed to be satisfied under process variation. Especially in path balancing, the perfect satisfaction of inequality (2.8) could easily be corrupted by a small variance of inertial gate delay. Hence the technique of path delay balancing is not effective and glitches cannot be completely suppressed under process variation.

Statistical delay modeling is introduced in [39] for gate sizing by non-linear programming. Gate delay is treated as a random variable with normal distribution. Hu [34-36] proposes a statistical path balancing approach by linear programming. The results show that power variation due to process variation can be reduced.

2.5 Summary

This chapter has introduced the field of low power design. Various techniques to reduce power consumption at the gate level are described. Dual-V_{th} assignment is a very efficient method to reduce the subthreshold leakage power. With process variation, subthreshold leakage increases exponentially, so a statistical approach is proposed to minimize the impact of process variation on leakage optimization. To reduce the unnecessary glitch power, hazard filtering and path balancing are used for glitch elimination. Although dynamic power is not sensitive to the process variation, the technique of path balancing becomes ineffective unless a statistical delay model is adopted to reflect the real conditions.

CHAPTER 3 DETERMINISTIC MILP FOR LEAKAGE AND GLITCH MINIMIZATION

The power dissipation of a CMOS circuit comprises dynamic power, short circuit power and static power. Leakage power is becoming a dominant contributor to the total power consumption with the continuing technology scaling. In the dynamic power, glitches as unnecessary signal transitions consume extra power. Compared to the other two power components, short circuit power can be ignored. In this chapter, we propose a mixed integer linear programming (MILP) formulation to globally minimize leakage power by dual-threshold design and eliminate glitches by path balancing.

3.1 Leakage and Delay

As discussed in Section 2.1.2, subthreshold leakage exponentially depends upon the threshold (V_{th}). Increasing V_{th} sharply reduces the subthreshold current, which is given by the following expression [42]:

$$I_{sub} = \mu_0 C_{ox} \frac{W}{L_{eff}} V_T^{\,2} e^{1.8} \exp\left(\frac{V_{gs} - V_{th}}{nV_T}\right) \cdot \left(1 - \exp\left(\frac{-V_{ds}}{V_T}\right)\right) \tag{3.1}$$

Spice simulation results on the leakage current of a two-input NAND gate are given in Table 3.1 for 70nm BPTM CMOS technology [1] (V_{dd} = 1V, Low V_{th} = 0.20V, High V_{th} =

0.32*V*). The leakage current of a high V_{th} gate is only about 2% of that of a low V_{th} gate. If all gates in a CMOS circuit could be assigned the high threshold voltage, the total leakage power consumed in the active and standby modes can be reduced by up to 98%, which is a significant improvement.

Table 3.1 Leakage currents for low and high V_{th} NAND gates.

Input vector	I_{leak} (nA)		
	Low V_{th}	*High* V_{th}	Reduction (%)
00	1.7360	0.0376	97.8
01	10.323	0.2306	97.8
10	15.111	0.3433	97.7
11	17.648	0.3169	98.2

However, according to the following equation, the gate delay increases with the increase of V_{th}.

$$T_{pd} \propto \frac{CV_{dd}}{(V_{dd} - V_{th})^{\alpha}} \tag{3.2}$$

where α equals 1.3 for short channel devices [82]. Table 3.2 gives the delays of a NAND gate obtained from Spice simulation when the output fans out to varying numbers of inverters. We observe that by increasing V_{th} form 0.20V to 0.32V, the gate delay increases by 30%-40%.

Table 3.2 Delays of low and high V_{th} NAND gates.

Number of fanouts	Gate delay (ps)		
	Low V_{th}	High V_{th}	% increase
1	14.947	21.150	41.5
2	22.111	30.214	36.6
3	29.533	39.171	32.6
4	37.073	48.649	31.2
5	44.623	58.466	31.0

We can make tradeoffs between leakage power and performance, leading to a significant reduction in the leakage power while sacrificing only some (or none) of circuit performance. Such a tradeoff is made in mixed integer linear programming (MILP). Results in Section 6.1.1 show that the leakage power of all ISCAS85 benchmark circuits can be reduced by over 90% if the delay of the critical path is allowed to increase by 25%.

3.2 A Deterministic MILP for Power Minimization

We use an MILP model to determine the optimal assignment of V_{th} while maintaining any given performance requirement on the overall circuit delay. To minimize the total leakage, the MILP assigns high V_{th} to the largest possible number of gates while controlling the critical path delays. Unlike the heuristic algorithms [45, 72, 96-101], MILP gives us a globally optimal solution as discussed in Section 3.4. To eliminate the glitch power, additional MILP constraints determine the positions and values of the delay elements to be inserted to balance path delays within the inertial delay of the incident gates. We can easily make a tradeoff between power reduction and performance degradation by changing the constraint for the maximum path delay in the MILP model.

3.2.1.1 Variables

Each gate is characterized by four variables:

***X_i*:** assignment of low or high V_{th} to gate i is specified by an integer X_i which can only be 0 or 1. A value 1 means that gate i is assigned low V_{th}, and 0 means that gate i is assigned high V_{th}. Each gate has two possible values of delays, D_{Li} and D_{Hi}, and two

possible values of leakages, I_{Li} and I_{Hi}, corresponding to low and high thresholds, respectively.

T_i: longest time gate *i* can take to produce an event after the occurrence of an input event at primary inputs of the circuit.

t_i: earliest time at which the output of gate *i* can produce an event after the occurrence of an input event at primary inputs of the circuit.

$\Delta d_{i,j}$: delay of a possible delay element that may be inserted at the input of gate *i* on the path from the output of gate *j*.

Thus, an *n*-input gate is characterized by *n*+7 quantities, i.e., *n* input buffer delay variables, two inertial delay constants, two leakage current values, one (0,1) integer variable, and two output timing window variables.

3.2.2 Objective Function

The objective function for the MILP is to minimize the sum of all gate leakage currents I_{leaki} and the sum of all inserted delays:

$$Min\left\{\sum_i I_{leaki} + \sum_i \sum_j \Delta d_{i,j}\right\} = Min\left\{\sum_i \left[X_i I_{Li} + (1 - X_i) I_{Hi}\right] + \sum_i \sum_j \Delta d_{i,j}\right\} \quad (3.4)$$

For a static CMOS circuit, the leakage power is

$$P_{leak} = V_{dd} \sum_i I_{leaki} \quad (3.5)$$

If we know the leakage currents of all gates, the leakage power can be easily obtained. Therefore, the first term in the objective functions of this MILP minimizes the sum of all gate leakage currents, i.e.,

$$Min \sum_i \left(X_i \cdot I_{Li} + (1 - X_i) \cdot I_{Hi} \right) \tag{3.6}$$

I_{Li} and I_{Hi} are the leakage currents of gate i with low V_{th} and high V_{th}, respectively. Recognizing that the subthreshold current of a gate depends on its input state, we make a leakage current look-up table of I_{Li} and I_{Hi} for all gates i using Spice simulation. These look-up tables are similar to Table 3.1 and are used for leakage power estimation. For the MILP, we need one set of I_{Li} and I_{Hi} for each gate and the average values from the look-up tables can be used.

Besides the leakage power, we minimize the glitch power, simultaneously. We insert minimal delays to satisfy the glitch elimination conditions at all gates. This leads to the second term in the objective function:

$$Min \sum_i \sum_j \Delta d_{i,j} \tag{3.7}$$

When implementing these delay elements, we use transmission gates with only the gate leakage (see Section 3.3).

The two terms in the objective function, ΣI_{leaki} and $\Sigma\Sigma\Delta d_{i,j}$, have different units and numerically ΣI_{leaki} is 50 to 1000 times larger than $\Sigma\Sigma\Delta d_{i,j}$ in our examples of benchmark circuits. Therefore, the objective function of Equation (3.4) puts greater emphasis on leakage power, assuming it to be the dominant contributor to the total power. Experimental results show that, when A is a very large positive constant and B equals to 1, the objective function Min ($A \cdot \Sigma I_{leaki} + B \cdot \Sigma\Sigma\Delta d_{i,j}$) generates results numerically identical to those obtained by the objective function of Equation (3.4) in which the terms are left

unweighted. In general, suitable weight factors *A* and *B* can be used to make tradeoffs between leakage power reduction and glitch power elimination.

3.2.3 Constraints

Constraints are imposed on each gate *i* with respect to each of its fanin *j*, where *j* refers to the gate providing the fanin:

$$T_i \geq T_j + \Delta d_{i,j} + \left[X_i \cdot D_{Li} + (1 - X_i) \cdot D_{Hi}\right] \tag{3.8}$$

$$t_i \leq t_j + \Delta d_{i,j} + \left[X_i \cdot D_{Li} + (1 - X_i) \cdot D_{Hi}\right] \tag{3.9}$$

$$X_i \cdot D_{Li} + (1 - X_i) \cdot D_{Hi} \geq T_i - t_i \tag{3.10}$$

where D_{Hi} and D_{Li} are the delays of gate *i* with high V_{th} and low V_{th}, respectively. With the increase in fanouts, the delay of the gate increases proportionately. Therefore, a look-up table is constructed using Spice simulation and specifies the delays for all gate types with varying number of fanouts. D_{Hi} and D_{Hi} for gate *i* are obtained from the look-up table whose entries are indexed by the gate type and the number of fanouts. Constraints (3.8), (3.9) and (3.10) ensure that the inertial delay of gate *i* is always larger than the delay difference of its input paths. This would be done by inserting the minimal number of delay elements while maintaining the critical path delay constraints.

We explain constraints (3.8), (3.9) and (3.10) using the circuit shown in Figure 3.1. Here the numbers on gates are gate numbers and not the delays. Bold lines show critical paths and two grey shaded triangles are delay elements possibly inserted on the input paths of gate 2. Similar delay elements are placed on all primary inputs and fanout branches throughout the circuit.

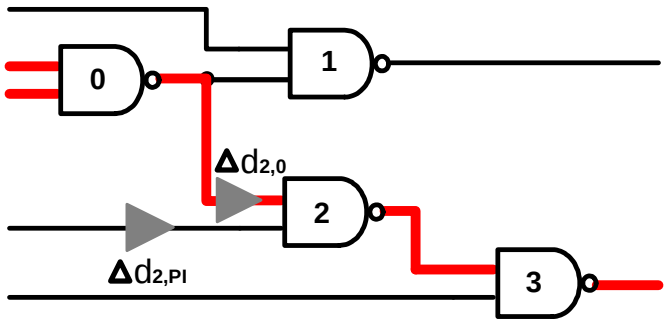


Figure 3.1 Circuit for explaining MILP constraints.

Let us assume that all primary input (PI) signals on the left arrive at the same time. For gate 2, one input is from gate 0 and the other input is directly from a PI. Its constraints corresponding to inequalities (3.8), (3.9) and (3.10) are:

$$T_2 \geq T_0 + \Delta d_{2,0} + [X_2 \cdot D_{L2} + (1 - X_2) \cdot D_{H2}] \tag{3.11}$$

$$T_2 \geq 0 + \Delta d_{2,PI} + [X_2 \cdot D_{L2} + (1 - X_2) \cdot D_{H2}] \tag{3.12}$$

$$t_2 \leq t_0 + \Delta d_{2,0} + [X_2 \cdot D_{L2} + (1 - X_2) \cdot D_{H2}] \tag{3.13}$$

$$t_2 \leq 0 + \Delta d_{2,PI} + [X_2 \cdot D_{L2} + (1 - X_2) \cdot D_{H2}] \tag{3.14}$$

$$[X_2 \cdot D_{L2} + (1 - X_2) \cdot D_{H2}] \geq T_2 - t_2 \tag{3.15}$$

The variable T_2 that satisfies inequalities (3.11) and (3.12) is the latest time at which an event (signal change) could occur at the output of gate 2. The variable t_2 is the earliest time at which an event could occur at the output of gate 2, and it satisfies both inequalities (3.13) and (3.14). Constraint (3.15) means that the difference of T_2 and t_2, which equals the delay difference between two input paths, is smaller than gate 2's inertial delay, which may be either the low V_{th} gate delay, D_{L2}, or the high V_{th} gate delay, D_{H2}.

The critical path delay T_{max} is specified at primary output (PO) gates 1 and 3, as:

$$T_i \le T_{\max} \ , \quad i = 1, 3 \tag{3.16}$$

T_{max} can be the maximum delay specified by the circuit designer. Alternatively, the delay of the critical path (T_c) can be obtained from a linear program (LP) by assigning all gates to low V_{th}, i.e., $X_i = 1$ for all i. The objective function of this LP minimizes the sum of T_k's where k refers to primary outputs. The critical path delay T_c is then the maximum of T_k's found by the LP.

If T_{max} equals to T_c, the actual objective function of the MILP model will be to minimize the total leakage current without affecting the circuit performance. By making T_{max} larger than T_c, we can further reduce leakage power with some performance compromise, and thus make a tradeoff between leakage power consumption and performance.

When we use this MILP model to simultaneously minimize leakage power with dual-V_{th} assignments and reduce dynamic power by balancing path delays with inserted delay elements, the optimized version for the circuit of Figure 3.2(a) is shown in Figure 3.2(b). In these figures the labels in or near gates are inertial delays.

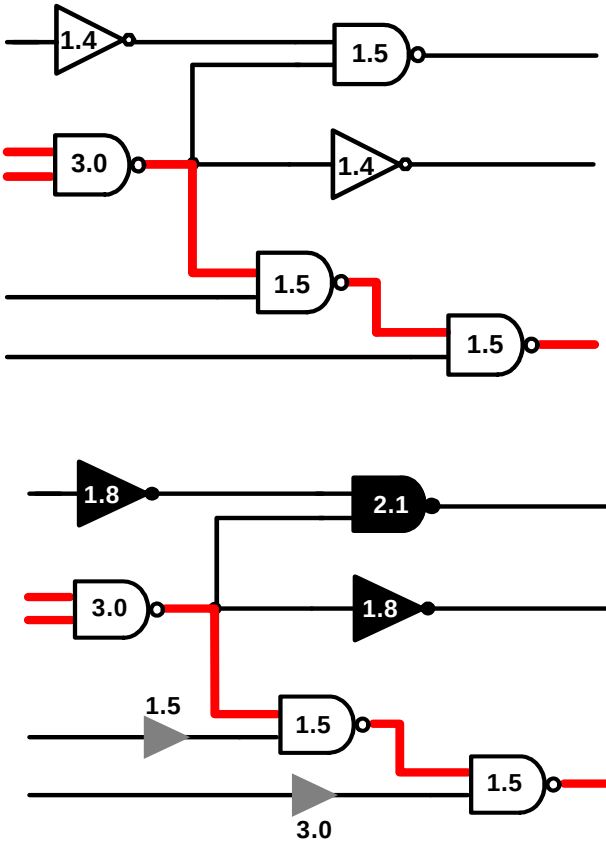


Figure 3.2 (a) An unoptimized circuit with high leakage and potential glitches, and (b) its corresponding optimized glitch-free circuit with low leakage.

Three black shaded gates are assigned high V_{th}. They are not on critical paths (shown by bold lines) and their delay increases do not affect the critical path delay. Although delay elements were assumed to be present on all primary inputs and fanout branches, only two were assigned non-zero values. They are shown as grey triangles with delays of 1.5 and 3.0 units, respectively. To minimize the additional leakage and dynamic power consumed by these delay elements, we implement them by CMOS transmission gates. In Section 3.3, we will show that an always-turned-on CMOS transmission gate can be used as a zero-subthreshold leakage and low-dynamic-power-consumption delay element [75-77].

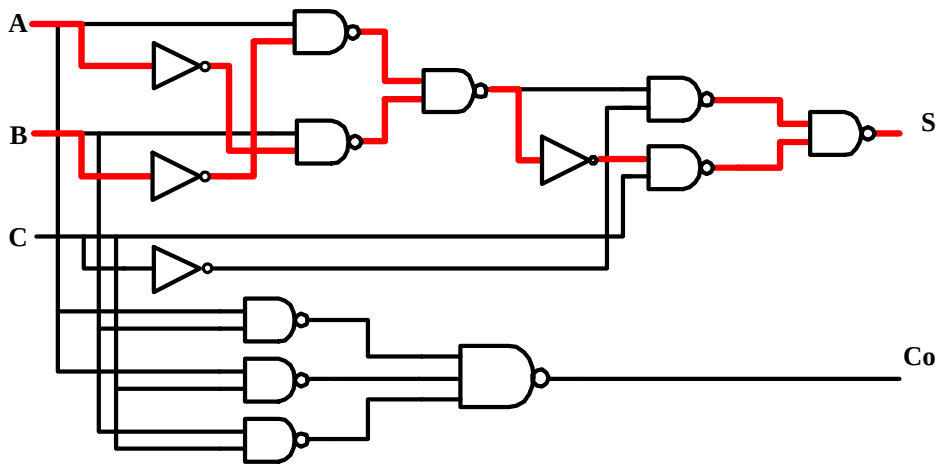


Figure 3.3 A full adder circuit with all gates assigned low V_{th} **(I_{leak} = 161 *nA*).**

A 14-gate full adder is used as a further illustration. Figure 3.3 is the original circuit with all low V_{th} gates. Critical paths are shown in bold lines. Figure 3.4(a) shows an MILP solution. All gates on non-critical paths were assigned high V_{th} (black shaded) to minimize leakage power. At the same time, three delay elements (grey shaded) are inserted to balance path delay to eliminate glitches. When the critical path delay is increased by 25%, the MILP gives the solution of Figure 3.4(b). Greater leakage power saving is achieved since some gates on the critical path are also assigned high V_{th}. All three circuits were implemented in the 70nm BPTM CMOS technology [1] we mentioned in Section 3.1. The three delay elements use high-V_{th} devices and their design is described in the next section. The leakage currents for the circuits of Figures 3.3 (unoptimized), 3.4(a) (optimized with no critical path delay increase) and 3.4(b) (optimized with 25% increase in critical path delay) were 161nA, 73nA and 16nA, respectively.

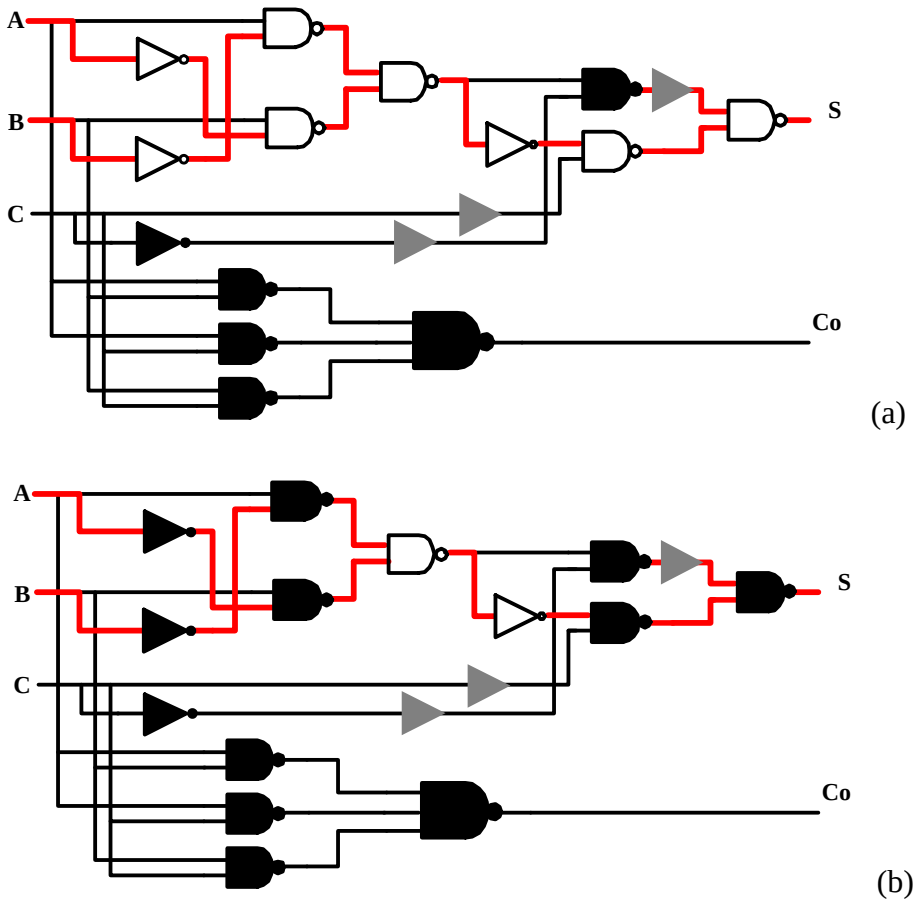


Figure 3.4 (a) Dual-V_{th} assignment and delay element insertion for $T_{max} = T_c$. **(I_{leak} = 73 *nA*), and** (b) Dual-V_{th} assignment and delay element insertion for $T_{max} = 1.25T_c$. **(I_{leak} = 16 *nA*)**

3.3 Delay Element Implementation

In our design, all delay elements are implemented by transmission gates, whose obvious advantage is that they consume very little dynamic power because they are not driven by any supply rails [60]. They also have lower area overhead and leakage power consumption compared with the more conventional two-cascaded-inverter buffer [75, 77, 92, 93]. CMOS transmission gates are adopted in our design to avoid the voltage drop when signal passes through series transistors.

3.3.1 Delay Element Comparison

The circuits in Figure 3.5, simulated for the subthreshold current by Spice, were used to compare the leakage power dissipation in the two delay elements. In Figure 3.5(a), there are only gate leakage paths and no subthreshold leakage since the two transistors are always turned on. In two cascaded inverters of Figure 3.5(b), besides gate leakage, subthreshold paths always exist. Hence, we can treat a transmission-gate delay element as a zero-subthreshold-leakage delay element.

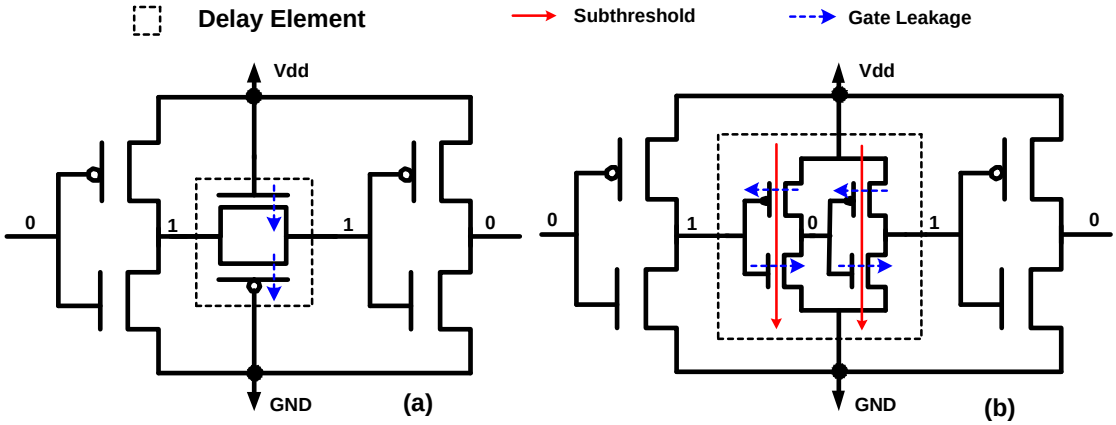


Figure 3.5 Delay elements: (a) CMOS transmission gate and (b) Cascaded inverters.

The delay of a transmission gate is given by [60]:

$$t_p = \ln(2) R_{eq} C_L \tag{3.17}$$

Where R_{eq} is the equivalent resistance of the CMOS transmission gate, and C_L is the load capacitance. By changing the widths and lengths of the transistors, we can change the delay of the transmission gate. We simulated the circuit of Figure 3.5(a) for nearly 80 transmission gates with transistors whose dimensions were varied. By subtracting the

delay of the circuit in which the transmission gate was replaced by a short, we obtained the delay of the transmission gate. These data were arranged in a look-up table of delays versus transmission gate dimensions. For any required delay between two entries in the look-up table, the size of the transmission gate is determined by interpolation.

The transmission gate delay elements avoid the comparatively larger capacitive dissipation and subthreshold leakage inherent in the alternative design of two inverter type of delay elements. However, the gate leakage of the transmission-gate delay element could become a concern, and will require further investigation.

3.3.2 Capacitances of a Transmission-Gate Delay Element

The purpose of path balancing by inserting delay elements is to eliminate glitches, so the capacitances contributed by transmission-gate delay elements should be considered carefully to calculate the extra dynamic switching power consumed by these delay elements. Figure 3.6 [73] shows the capacitances in a CMOS transistor, including diffusion capacitances, channel capacitances, and structure overlap capacitances.

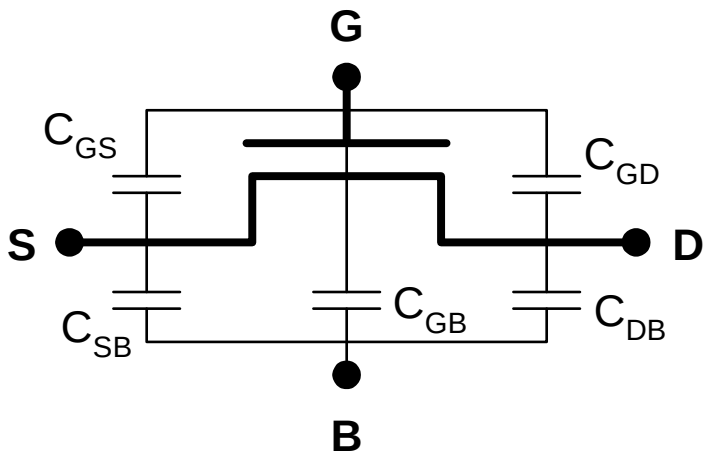


Figure 3.6 Capacitances in a MOS transistor.

Diffusion Capacitances, C_{SB} and $C_{DB,}$ are contributed by reversed-biased source-bulk and drain-bulk PN junctions. Their values are determined by the following the equation [73]:

$$C_{diff} = C_j L_s W + C_{jsw}(2L_s + W) \tag{3.18}$$

where C_j is the junction capacitance per unit area and C_{jsw} is the junction capacitance per unit perimeter. L_s is the length of the PN junction and it is usually two or three times the minimum channel length. W is the channel width.

Channel Capacitances, C_{gb}, C_{gs} and C_{gd} are capacitances between the gate and the bulk, the source, the drain regions. Although their values depend on the operation region of the transistor, we can use Equation (3.19) to estimate their total value [73]:

$$C_{ch} = C_{gb} + C_{gs} + G_{gd} = C_{ox} W L_{eff} \tag{3.19}$$

where C_{ox} is the oxide capacitance per unit area, and L_{eff} is the effective channel length.

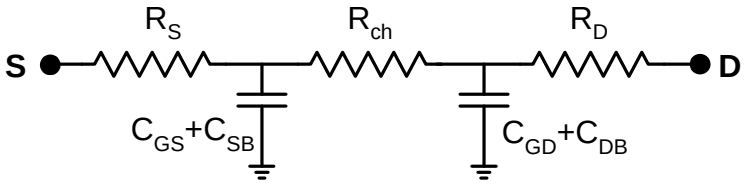
Overlap Capacitances are caused by the lateral diffusion in which source and drain region extend somewhat below gate oxide with the length of X_d [73]. Equation (3.20) gives their expression.

$$C_{gsO} = C_{gdO} = C_{ox} W X_d \tag{3.20}$$

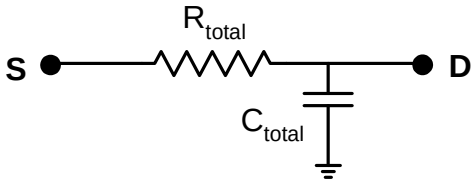
In Figure 3.6, $C_{GB} = C_{gb}$, $C_{GS} = C_{gs} + C_{gsO}$ and $C_{GD} = C_{gd} + C_{gdO}$.

As a delay element used for path balancing, a NMOS transmission gate is always turned on, so C_{GB} is zero. Since "G", as shown in Figure 3.6, is virtually connected to the ground, during the transition of the input signal, the distributed and lumped RC models of a NMOS transmission gate can be as shown in Figure 3.7(a) and Figure 3.7(b), respectively. Our design uses the lumped RC model to estimate the largest possible

dynamic power overhead contributed by the transmission-gate delay elements. The capacitance of a CMOS transmission gate is exactly twice that of an NMOS transmission gate since NMOS and PMOS transistors usually have the same sizes in a CMOS transmission gate.



(a) distributed RC model



$R_{total}=R_{ch}+R_S+R_D \quad C_{total}=C_{GS}+C_{GD}+C_{SB}+C_{DB}$

(b) lumped RC model

Figure 3.7 (a) Distributed and (b) Lumped RC models of a NMOS transmission gate.

From Equation (3.18), we see that diffusion capacitances depend upon the width *W* of the transmission gate, but not on the effective channel length L_{eff}. To minimize diffusion capacitances, we implement all transmission-gate delay elements with the minimal width but longer channel transistors, which are in contrast to the cells in the standard cell library whose length is usually kept minimum to achieve the fastest speed.

$$
\begin{aligned}
t_p &= \ln(2) R_{eq} C_L \\
&\propto \frac{L}{W}(C_{trans\_total} + C_{load\_chan}) \\
&= \frac{L}{W}\{(a \cdot LW + b \cdot W + c) + C_{load\_chan}\} \\
&= L(a \cdot L + b + \frac{c + C_{load\_chan}}{W})
\end{aligned}
\tag{3.21}
$$

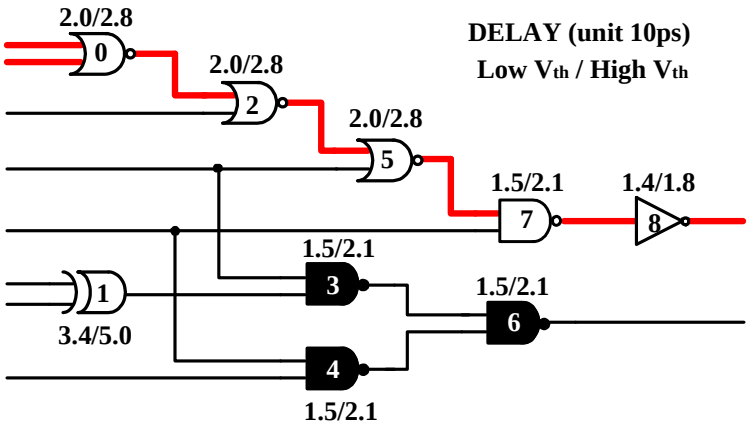
In Equation (3.21), the loading capacitance (C_L) of a transmission gate is comprised of two components, the total capacitance (C_{trans_total}) of the transmission gate and the channel capacitance (C_{load_chan}) of the gate driven by this transmission gate. We re-express C_{trans_total} as $a \cdot LW + b \cdot W + c$ according to Equations (3.18-3.20), where a, b and c are constants specified for the technology, and W and L are the width and length of the transmission gate, respectively. Equation (3.21) shows that to implement a certain delay, the smallest L is needed with minimum W. This surely reduces the channel capacitance of the transmission gate that is proportional to $L \cdot W$. In other words, a minimal-width transmission gate can guarantee the minimum C_{total} and causes the smallest dynamic power overhead.

3.4 MILP and Heuristic Algorithms

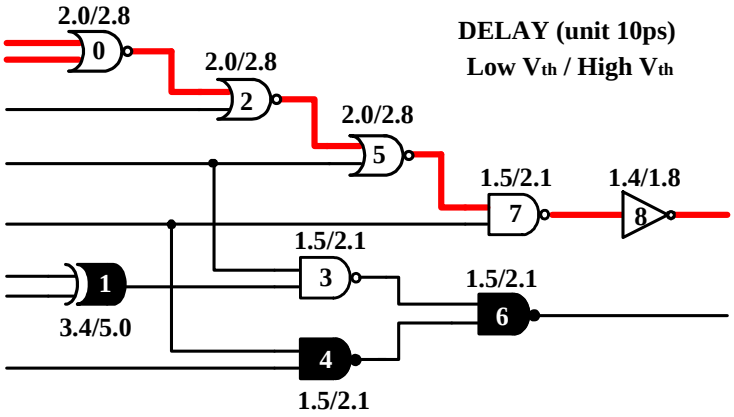
In Section 2.2.1, we mentioned several heuristic algorithms [45, 72, 96-101] used for dual-V_{th} assignment. Heuristic algorithms normally aim at achieving a locally optimal solution. In an MILP, both the objective function and constraints are linear. The linear objective function is definitely convex. The feasible region is also convex since the linear constraints geometrically define a convex polyhedron. Therefore, according to Karush-Kuhn-Tucker conditions [50], all locally optimal solutions are also globally optimal.

Thus, our MILP formulation always ensures a globally optimal solution. But the run time to solve a linear programming problem, especially an integer linear programming problem, can be a concern since in the worst case the run time exponentially depends on the number of constraints and variables. This will be discussed in Section 6.3.

To illustrate the point, we examine the *backtracking algorithm* [98] as an example to show the advantage of the MILP. In Figure 3.8, the XOR gate (gate 1) close to the primary inputs has the largest leakage power reduction if assigned a high threshold. However, in Figure 3.8(a), the slacks for the non-critical paths are first consumed by gates 6, 3 and 4, which are closer to primary outputs. Hence, by the time the backtracking arrives at the XOR gate the slack has already been used up and it cannot be assigned high-V_{th}. In Figure 3.8(b), MILP considers leakage reduction and delay increase of each gate simultaneously, making sure that the best candidates (gates with the largest leakage reduction without violating the timing constrains) are selected. Due to the global optimization, the MILP achieves 26% greater leakage power saving compared to the heuristic backtracking algorithms. Other heuristic algorithms have similar problems, because the available slack for each gate must depend on the search direction or the selected cut [96] in the circuit graph. Thus, a global optimization cannot be guaranteed.



(a) Backtracking algorithm: optimized leakage current is 79.2nA.



(b) MILP: optimized leakage current is 58.1nA

Figure 3.8 Comparison of MILP with heuristic backtracking algorithm.

3.5 Summary

In this chapter an MILP formulation to optimize the total power consumption by dual-V_{th} assignment and path balancing is proposed. It is a deterministic technique in which inertial delays of gates are assumed to have fixed values. However, variations of process parameters, especially in nanometer technologies, can change gate delays and affect the path delay balancing, causing incomplete suppression of glitches. Hu and

Agrawal [34, 37] propose a statistical analysis to treat the gate delays as random variables with normal distributions. The results show that the power distribution due to the process variation can be reduced. Our deterministic MILP models can also be extended as statistical MILP models to minimize the impact of the process variation on the leakage power optimization and glitch elimination. These will be discussed in Chapters 4 and 5, respectively.

CHAPTER 4 STATISTICAL MILP FOR LEAKAGE OPTIMIZATION UNDER PROCESS VARIATION

The increased variation of process parameters of nanoscale devices not only results in a higher average (mean) leakage but also causes a larger spread (standard variation) of leakage power [88]. In [17], twenty times difference in leakage and thirty percent variation in performance are observed. Some low leakage chips with very slow speeds and some other faster but very leaky chips have to be discarded. Therefore, both power yield and timing yield are seriously affected by the process variation. In this chapter, we propose a statistical MILP formulation to optimize the leakage power considering process variation. Results show that both mean and standard deviation of the leakage power distribution are reduced by this statistical method, compared to the deterministic approach discussed in Chapter 3.

4.1 Effects of Process Variation on Leakage Power

Process variations are basically separated into inter-die and intra-die variations. Inter-die variation or global variation refers to variation from wafer to wafer, or die to die on a same wafer, while intra-die variation, or local variation, occurs across an individual die. That means that on the same chip, devices at different locations may have different process parameters. Since inter-die variation affects all the devices on a chip in the same way, it has a stronger effect on power and performance.

Channel length, doping concentration and oxide thickness are the most important variations in devices. Oxide thickness is well controlled and generally only its inter-die variation is considered. Its effect on performance and power are often lumped into the channel length variation. Channel length variations are caused by photolithography proximity effects and deviation in the optics. Threshold voltages vary due to different doping concentration and annealing effects, mobile charge in the gate oxide, and discrete dopant variations caused by the small number of dopant atoms in tiny transistors [102].

Table 4.1 Leakage power distribution of un-optimized C432 benchmark circuit under local effective gate length variation.

L_{eff} variation	Nominal (μW)	Mean (μW)	S.D. (μW)	S.D. / mean	(mean-nominal)/nominal
10%	2.60E-06	2.75E-06	8.55E-08	3.10%	6.06%
20%	2.60E-06	3.39E-06	2.97E-07	8.75%	30.71%
30%	2.60E-06	5.53E-06	1.39E-06	25.17%	112.86%

Due to the exponential relation of leakage current with process parameters, such as the effective gate length, oxide thickness and doping concentration, process variations can cause a significant increase in the leakage current. Gate leakage is most sensitive to the variation in oxide thickness (T_{ox}), while the subthreshold current is extremely sensitive to the variation in effective gate length (L_{eff}), oxide thickness (T_{ox}) and doping concentration (N_{dop}). Twenty percent variations in effective channel length and oxide thickness can cause up to 13 and 15 times differences, respectively, in the amount of subthreshold leakage current. Gate leakage can have 8 times difference due to a 20% variation in oxide thickness. Compared with the gate leakage, the subthreshold leakage is more sensitive to parameter variations [66].

Subthreshold leakage depends exponentially on several key process parameters, V_{th}, L_{eff} and T_{ox}. The variation in these process parameters will surely cause both an increase of the average value and a large spread in the leakage power. Simulation results in Table 4.1 show that there is a 112.9% of average power increase and a 25.2% standard deviation when 30% (3σ) of local L_{eff} variation is applied to the un-optimized C432. From Figure 4.1, it is remarkable that process variation significantly affects the leakage power. With the increase of the process variation, the spread of the leakage power becomes wider and wider, and the average leakage also has a huge increase.

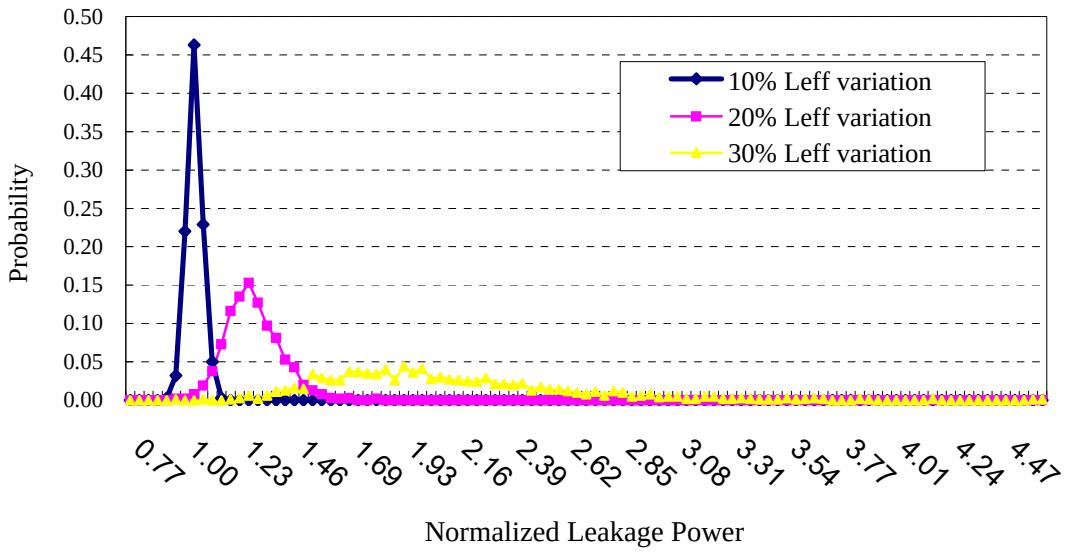


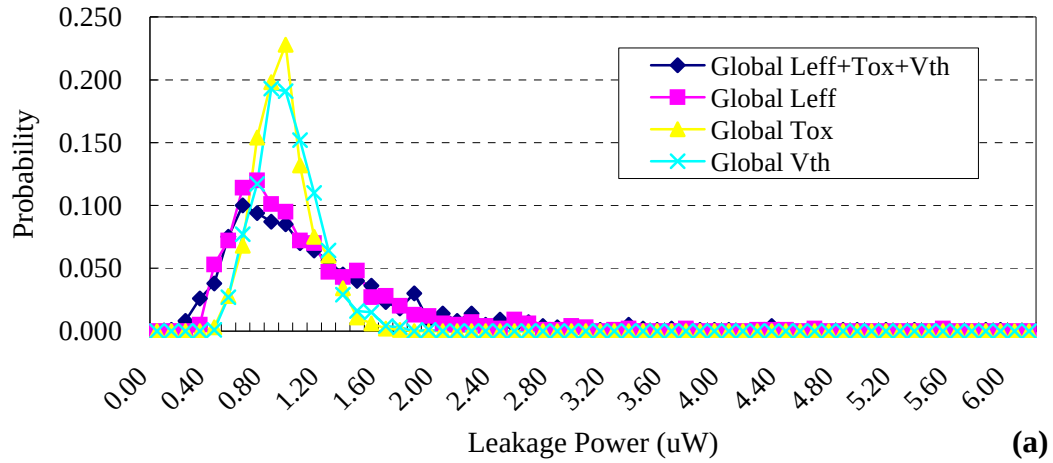
Figure 4.1 Leakage power distribution of un-optimized C432 under local effective gate length variation.

In Chapter 3, leakage power of ISCAS'85 benchmark circuits is optimized by a deterministic MILP formulation that ignores the effect of process variations. Table 4.2 compares the subthreshold leakage power distributions of the optimized dual-V_{th} C432 under different process parameter variations.

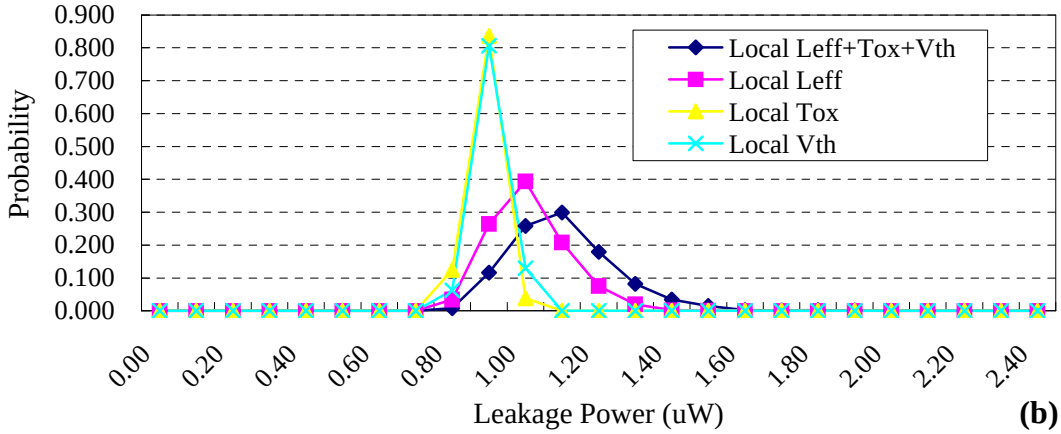
Table 4.2 Comparison of leakage power in the deterministically optimized dual-V_{th} C432.

process parameter (3σ=15%)		nominal (nW)	mean (nW)	S.D. (nW)	S.D. / mean	(mean-nominal) / nominal	max dev. from nominal (nW)	max dev / nominal
L_{eff}	local	906.9	1059	103.6	9.8%	16.8%	611.6	67.4%
	global	906.9	1089	599.1	55.0%	20.1%	4652.0	513.0%
T_{ox}	local	906.9	939.6	33.7	3.6%	3.6%	136.9	15.1%
	global	906.9	938.6	199.9	21.3%	3.5%	795.8	87.7%
V_{th}	local	906.9	956.7	36.4	3.8%	5.5%	171.0	18.9%
	global	906.9	964.4	219.8	22.8%	6.3%	1028.0	113.4%
L_{eff} + T_{ox} + V_{th}	local	906.9	1155	140.8	12.2%	27.4%	1044.0	115.1%
	global	906.9	1164	719.4	61.8%	28.3%	5040.0	555.7%

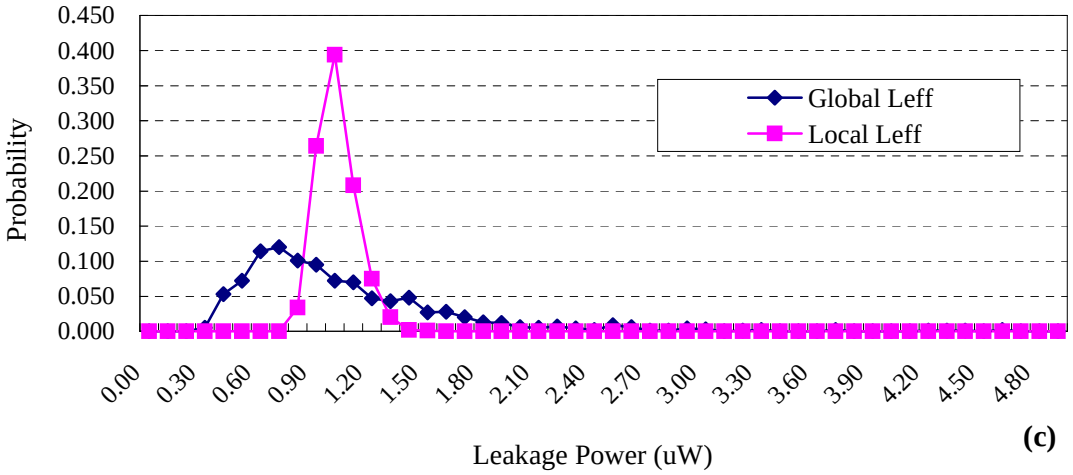
Table 4.2, and Figures 4.2(a) and 4.2(b) show that among T_{ox}, L_{eff} and V_{th}, subthreshold leakage is most sensitive to the variation in L_{eff}. The simulation results are consistent with those in [66, 87]. Table 4.2, Figure 4.2(c) and Figure 4.2(d) demonstrate that leakage has a stronger influence of global variation than that of local variation. The reason is obvious. In global variation, the leakage power of all the devices on the same chip either increases or decreases in the same way which causes a wider leakage spread and larger mean, while local variation affects the devices across the chip randomly, and hence leads to a narrower spread and smaller mean. But irrespective of the type of variation, the mean of the leakage power always increases due to the exponential relation of the subthreshold leakage to some key process parameters.


0.250
0.200
0.150
0.100
0.050
0.000
Probability
Global Leff+Tox+Vth
Global Leff
Global Tox
Global Vth
0.00
0.40
0.80
1.20
1.60
2.00
2.40
2.80
3.20
3.60
4.00
4.40
4.80
5.20
5.60
6.00
Leakage Power (uW)

(a)


0.900
0.800
0.700
0.600
0.500
0.400
0.300
0.200
0.100
0.000
Probability
Local Leff+Tox+Vth
Local Leff
Local Tox
Local Vth
0.00
0.20
0.40
0.60
0.80
1.00
1.20
1.40
1.60
1.80
2.00
2.20
2.40
Leakage Power (uW)

(b)


0.450
0.400
0.350
0.300
0.250
0.200
0.150
0.100
0.050
0.000
Probability
Global Leff
Local Leff
0.00
0.30
0.60
0.90
1.20
1.50
1.80
2.10
2.40
2.70
3.00
3.30
3.60
3.90
4.20
4.50
4.80
Leakage Power (uW)

(c)

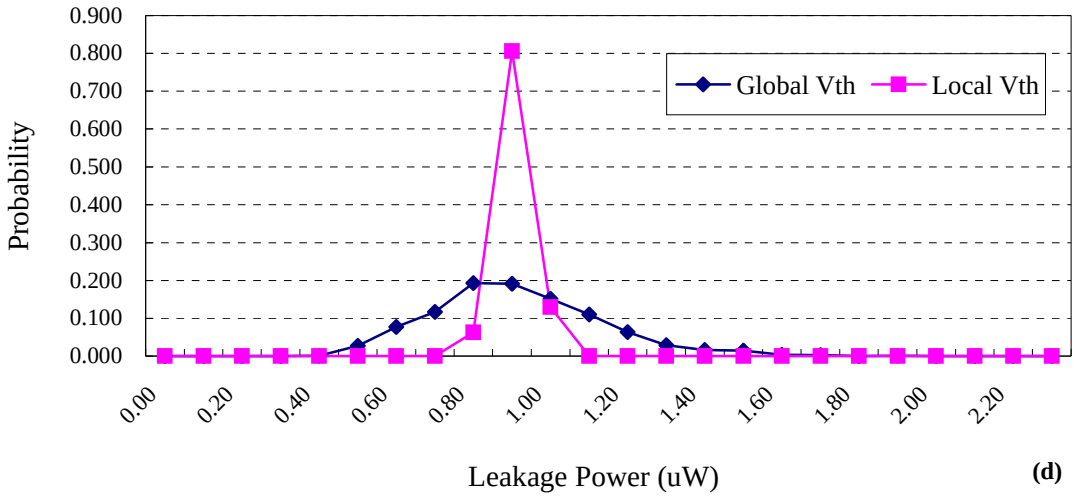


Figure 4.2 Leakage power distributions of the deterministically optimized dual-V_{th} C432 due to process parameter variations, (a) global variations, (b) local variations, (c) effective gate length variations, and (d) threshold voltage variations.

4.2 Overview of Deterministic Dual-V_{th} Assignment by MILP

In the deterministic approach proposed in Chapter 3, the delay and subthreshold current of every gate are assumed to be fixed and without any effect of the process parameter variation. Basically, such method can be divided into two groups: heuristic algorithms [45, 72, 96-101] and linear programming algorithms [31, 56, 58, 70]. Heuristic algorithms give a locally optimal solution while a linear programming formulation ensures a globally optimum solution.

An MILP that optimizes the leakage power and assigns dual-V_{th} to gates in one step [56-58] has an advantage over an iterative procedure [70], which must assume power-delay sensitivities to be constants in a small range. Figure 4.3 gives the basic idea of the MILP method [56, 58] that minimizes the total subthreshold leakage without compromising the circuit performance by dual-V_{th} assignment.

Minimize	$\sum_i I_{subnom,i}$	$\forall i \in$ *gate number*
Subject to	$T_{POk} \le T_{\max}$	$\forall k \in PO$

Figure 4.3 Basic idea of using MILP to optimize leakage.

A detailed version for the MILP formulation is presented in Figure 4.4. X_i is an integer that can only be either 0 or 1. A value 1 means that gate *i* is assigned low V_{th}, and 0 means that gate *i* is assigned high V_{th}. T_i is the latest arrival time at the output of gate *i*. Each gate in the design library has low and high threshold versions, which are characterized, using Spice simulation, for their leakage in various input states and gate delays, which also depend on the number of fanouts.

Minimize	$\sum_i \{X_i \cdot I_{subnom,L,i} + (1-X_i) \cdot I_{subnom,H,i}\}$	$\forall i$	(D-O)
Subject to			
	$D_i = X_i \cdot D_{nom,L,i} + (1-X_i) \cdot D_{nom,H,i}$	$\forall i$	(D-C1)
	$T_i \ge T_j + D_i$	$\forall j \in$ *fanin of gate i*	(D-C2)
	$X_i = 0 \ \ or \ \ 1$	$\forall i$	(D-C3)
	$T_{POk} \le T_{\max}$	$\forall k \in PO$	(D-C4)

Figure 4.4 Detailed deterministic MILP formulation for leakage minimization.

4.3 Statistical Dual-V_{th} Assignment

Process variations include inter-die and intra-die variations, or global and local variations. For inter-die variations, because the inertial gate delay of every device on the same die changes with the same percentage, the solutions for the deterministic and statistical approaches are exactly the same. Since our objective is to have a statistical

MILP formulation that enhances the deterministic approach to leakage optimization under process variations, we ignore the inter-die variation. In the remainder of this chapter, process variation will only mean intra-die variation.

Leakage current is composed of reverse biased PN junction leakage, gate leakage and subthreshold leakage. In a sub-micron process, PN junction leakage is much smaller than the other two components. Gate leakage is most sensitive to the variation in T_{ox}, and changes in the gate leakage due to other process parameter variations can be ignored [66]. Further, assuming T_{ox} to be a well-controlled process parameter [79, 81, 102], we ignore the gate leakage variation in our design, focusing only on changes in the subthreshold leakage due to process variation.

Subthreshold current depends exponentially on some key process parameters, such as, the effective gate length, oxide thickness and doping concentration. Process variation can severely affect both power and timing yields of a design optimized by a deterministic method. Because fixed subthreshold leakage and gate delay do not represent the real circuit condition, statistical modeling should be used. This is discussed next.

4.3.1 Statistical Subthreshold Leakage Modeling

Subthreshold current has an exponential relation with the threshold voltage, which in turn is a function of oxide thickness, effective channel length, doping concentration, etc. T_{ox} is a fairly well-controlled process parameter and does not significantly influence subthreshold leakage variation [79, 81, 102]. Therefore, we only consider variations in L_{eff} and N_{dop}.

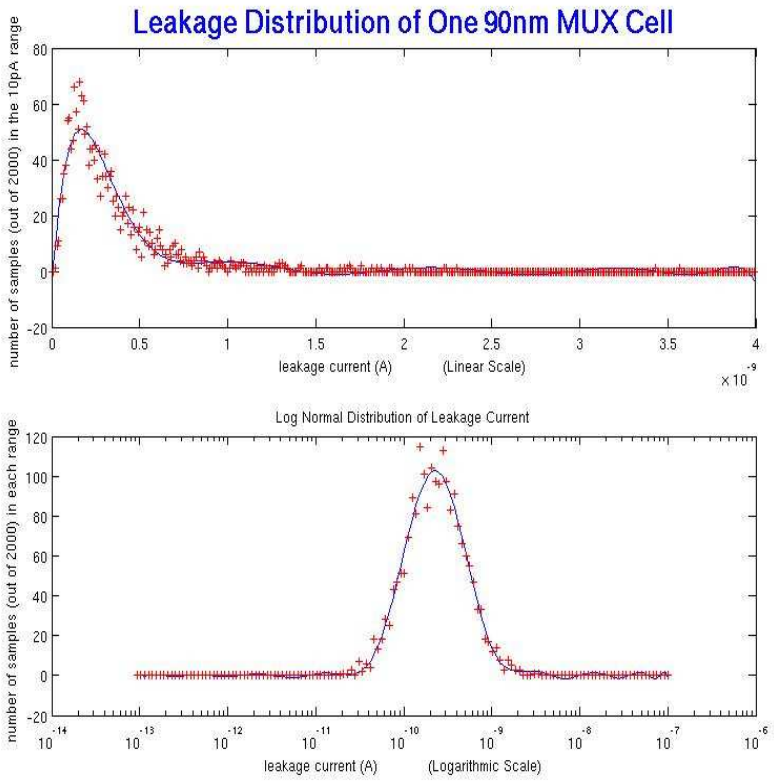


Figure 4.5 Monte Carlo Spice simulation for leakage distribution of a MUX cell in TSMC 90nm CMOS technology.

The statistical subthreshold model can be written as [79]:

$$I_{sub} = I_{sub,nom} \cdot \exp\left(- \frac{\Delta L_{eff} + c_2 \Delta L_{Leff}^{\ 2} + c_3 \Delta V_{th,Ndop}}{c_1} \right) \tag{4.1}$$

where, ΔL_{eff} is the change in the effective channel length due to the process variation and $\Delta V_{th,Ndop}$ is the change in the threshold voltage due to the random distribution of doping concentration, N_{dop}. Both are random variables with a normal (Gaussian) distribution, N(0,1). Fitting parameters c_1, c_2 and c_3 are determined from the Spice simulation. From equation (4.1), it is obvious that I_{sub} has a lognormal distribution. Figure 4.5 gives the leakage distribution of one multiplexer (MUX) cell in TSMC 90nm CMOS technology

obtained by Monte Carlo Spice simulation. The x-axes in the upper and lower figures have linear and logarithmic scales, respectively. It is clear from the bottom figure that subthreshold leakage current has a lognormal distribution.

The total leakage current of a circuit, which is the sum of subthreshold currents of individual gates, has an approximately lognormal distribution. Rao *et al.* [79] use the *central limit theorem* to estimate this lognormal distribution by its mean value with the assumption that there is a large number of gates in the circuit, which is indeed the case for most present day chips. Hence, the total leakage can be expressed as:

$$I_{sub,total} = \sum_i I_{sub,i} \cong E\left[\sum_i I_{sub,i}\right] = S_L \cdot S_V \cdot \sum_i I_{subnom,i} \tag{4.2}$$

where,

$$S_L = \frac{1}{\sqrt{1+\frac{2\lambda_2}{\lambda_1}\sigma^2_{\Delta L_{eff}}}} \cdot \exp\left(\frac{\sigma^2_{\Delta L_{eff}}}{2\lambda_1^2 + 4\sigma^2_{\Delta L_{eff}}\lambda_1\lambda_2}\right) \tag{4.3}$$

$$S_V = \exp\left(\frac{\lambda_3^2\sigma^2_{\Delta V_{th,Ndop}}}{2\lambda_1^2}\right) \tag{4.4}$$

S_L and S_V are scale factors introduced due to local variations in L_{eff} and $V_{th,Ndop.}$ λ_1, λ_2 and λ_3 are fitting parameters. For a given process, $\sigma_{\Delta Leff}$ and $\sigma_{\Delta Vth,Ndop}$ are predetermined. Therefore, in our statistical linear programming formulation, the objective function is a sum of all nominal (without the effect of process variation) subthreshold leakage currents, multiplied by scale factors, S_L and S_V.

4.3.2 Statistical Delay Modeling

The deterministic gate delay D is given by [82]:

$$D \propto \frac{CV_{dd}}{(V_{dd} - V_{th})^{\alpha}} \tag{4.5}$$

where α equals 1.3 for the short channel model. Similar to the subthreshold current model, the V_{th} deviation due to the process parameter variation is also a consideration in our statistical delay model. The change of V_{th} due to the variation of process parameters can be expressed as [26]:

$$V_{th} = V_{th0} - \sum_i \beta_{X_i} \frac{X_{i0} - X_i}{X_{i0}} \tag{4.6}$$

where X_i is a process parameter, X_{i0} is the nominal value of X_i, and β_{Xi} is a constant for the specific technology.

To get an approximated linear relation between D and the variations of the process parameters, equation (4.5) is expanded as a Taylor series (4.7) and only the first order term is retained because higher orders terms being relatively small can be ignored.

$$D_{X_1, X_2, \ldots} = D(X_{10,} X_{20,} \ldots) + \sum (X_{i0} - X_i) \frac{dD}{dX_i} |_{X_{i0}} \tag{4.7}$$

Let $\{X_1, X_2\} = \{L_{eff}, N_{dop}\}$. Combining equations (4.6) and (4.7), we get:

$$D_i = D_{nom,i} \left(1 + c_{i1} \frac{\Delta L_{eff}}{L_{eff\,0}} + c_{i2} \frac{\Delta N_{dop}}{N_{dop0}} \right) \tag{4.8}$$

where, c_{i1}, c_{i2} are sensitivities of gate delay with respect to the variation of each process parameter and can be obtained from Spice simulation. L_{eff} and N_{dop} are normal $N(0,1)$ random variables. Therefore, in the statistical analysis, D_i becomes a random variable, which also has a normal distribution. Let,

$$r_i = c_{i1}\frac{\Delta L_{eff}}{L_{eff\,0}} + c_{i2}\frac{\Delta N_{dop}}{N_{dop0}} \tag{4.9}$$

Equation (4.8) becomes:

$$D_i = D_{nom,i}(1 + r_i) \tag{4.10}$$

Since r_i is a random variable with with Gaussian distribution, $N(0, \sigma_r^2)$, μ_{Di}, the mean value of D_i, is equivalent to D_{nomi} , the nominal delay of gate i. The standard deviation of D_i is the same as that of r_i.

4.3.3 MILP for Statistical Dual-V_{th} Assignment

In the statistical approach to minimize leakage power by dual-V_{th} assignment (Figure 4.6 and Figure 4.7), the delay and subthreshold current are both random variables, and η is the expected timing yield. The power yield is not considered because in Section 6.2, we will find that the statistical approach can get about 30% additional leakage power reduction for most circuits compared to the deterministic approach.

Minimize	$I_{sub,total}$	(4.11)
Subject to	$P(T_{POi} \le T_{\max}) \ge \eta$	(4.12)

Figure 4.6 Basic MILP for statistical dual-V_{th} assignment.

In Figure 4.6, T_{POi} is the path delay from primary input to the i_{th} primary output and is assumed to have a normal (Gaussian) distribution $N(\mu_{TPOi},\sigma^2_{TPOi})$. Inequality (4.12) allows leakage to be optimized with timing yield η and it can be expressed in a linear format by the percent point function Φ^{-1} [62]:

$$\mu_{T_{PO_k}} + \sigma_{T_{PO_k}} \cdot \Phi^{-1}(\eta) \le T_{\max} \tag{4.13}$$

In statistical linear programming (Figure 4.7) all variables, except X_i, are random variables with normal distributions.

Minimize

$$S_L \cdot S_V \cdot \sum_i I_{subnom,i} = S_L \cdot S_V \cdot \sum_i \left\{ X_i \cdot I_{subnom,L,i} + (1 - X_i) \cdot I_{subnom,H,i} \right\}$$

$$\forall i \in \textit{gate number} \qquad \text{(S-O)}$$

Subject to

$\forall i \in \textit{gate number}$

$$\mu_{Di} = X_i \cdot D_{nom,L,i} + (1 - X_i) \cdot D_{nom,H,i} \qquad \text{(S-C1)}$$

$$\sigma_{Di} = \sigma_r \cdot \mu_{Di} \qquad \text{(S-C2)}$$

$$\mu_{Ti} \ge \mu_{Tj} + \mu_{Di} \qquad \forall j \in \textit{fanin of gate } i \qquad \text{(S-C3)}$$

$$\sigma_{Tj,Di} = k(\sigma_{Tj} + \sigma_{Di}) \qquad \text{(S-C4)}$$

$$temp_{Ti} \ge \mu_{Tj} + \mu_{Di} + 3\sigma_{Tj,Di} \qquad \text{(S-C5)}$$

$$\sigma_{Ti} = (temp_{Ti} - \mu_{Ti})/3 \qquad \text{(S-C6)}$$

$$X_i = 0 \quad or \quad 1 \qquad \text{(S-C7)}$$

$\forall k \in PO$

$$\mu_{T_{PO_k}} + \sigma_{T_{PO_k}} \cdot \Phi^{-1}(\eta) \le T_{\max} \qquad \text{(S-C8)}$$

Figure 4.7 Detailed formulation of statistical dual-V_{th} assignment MILP.

Comparing the deterministic MILP (Figure 4.4) with the statistical MILP (Figure 4.7), we observe the following differences:

- The deterministic gate delay in (D-C1) is extended to (S-C1) and (S-C2) to get the mean and standard deviation of the statistical delay.
- (D-C2) is extended to (S-C3) through (S-C6) to get the mean and standard deviation of the statistical arrival time T_i at the output of gate *i*. Section 4.4 briefely explains the transfer from (D-C2) to (S-C3) through (S-C6).
- (D-C4) is updated to (S-C8) to ensure specified timing yield under process variation.

4.4 Linear Approximations

In linear programming, all the expressions and constraints should be linear functions. However, in statistical analysis, some nonlinear operations are present. We, therefore, use the following linear approximations.

- **ADD, A = B + C**

If *B* and *C* are $N(\mu, \sigma^2)$ random variables, then their sum *A* also has a normal distribution. 'Add' is a linear function, but in statistical analysis, to obtain the standard deviation σ_A, we must deal with $\sigma_A^2 = \sigma_B^2 + \sigma_C^2$, which is a nonlinear operation. Considering,

$$\frac{(\sigma_B + \sigma_C)^2}{2} \le \sigma_B^2 + \sigma_C^2 \le (\sigma_B + \sigma_C)^2 \tag{4.14}$$

One can find a linear approximation [34, 37]:

$$\sigma_A = \sqrt{\sigma_B^2 + \sigma_C^2} = k(\sigma_B + \sigma_C) \quad \text{with } k \in [\frac{\sqrt{2}}{2}, 1] \tag{4.15}$$

The optimal solution partially depends upon the value of *k*. If we let *k* be 1, then the standard deviation of a random variable obtained by Equation (4.15) is probably larger than its real value and hence the constraints in the statistical MILP formulation (Figure 4.7) are too tight to get a feasible solution. On the contrary, the smallest *k*, ~0.707, is too optimistic for the real condition. Hence, we let *k* take a middle value 0.85.

- **MAX, A = MAX(B, C)**

If *B* and *C* are N(μ,σ^2) random variables, *A* does not necessarily have a normal distribution [56, 58]. However, a normal linear approximation with following mean and standard deviation has been used [34, 37]:

$$\mu_A = \max(\mu_B, \mu_C) \tag{4.16}$$

$$\sigma_A = \left\{\max(\mu_B + 3\sigma_B, \mu_C + 3\sigma_C) - \mu_A\right\}/3 \tag{4.17}$$

The error in this approximation has been shown to be small [34, 37].

- **MIN, A = MIN(B, C)**

Similarly, for function *A* = *min*(*B*,*C*), we use Equations (4.18) and (4.19) to estimate the mean and standard deviation of random variable *A*.

$$\mu_A = \min(\mu_B, \mu_C) \tag{4.18}$$

$$\sigma_A = \left\{\mu_A - \min(\mu_B - 3\sigma_B, \mu_C - 3\sigma_C)\right\}/3 \tag{4.19}$$

The above linear approximations are used in our statistical analysis to model the leakage optimization problem under process variation by a linear programming formulation proposed in this chapter and in Chapter 5.

4.5 Summary

The increased process parameter variations in nanoscale devices lead to a large spread of leakage power distribution and a higher average leakage. In this chapter, we propose an MILP formulation for leakage optimization by dual-V_{th} assignment in a statistical sense. Compared to the deterministic approach, which has to analyze the worst case to consider the process variation, statistical MILP formulation has a more flexible optimization space and can obtain an optimized dual-threshold circuit with less sensitivity to the process variation.

CHAPTER 5 TOTAL POWER MINIMIZATION WITH PROCESS VARIATION BY DUAL-THRESHOLD DESIGN, PATH BALANCING AND GATE SIZING

Compared to subthreshold leakage, dynamic power is normally much less sensitive to the process variation due to its approximately linear relation to the process parameters. However, the deterministic technique discussed in Chapter 3, which uses path balancing to eliminate glitches, becomes somewhat ineffective under process variation because the perfect hazard filtering conditions can easily be corrupted with a very slight variation in some process parameters. The average dynamic power of a circuit optimized by the deterministic path balancing approach increases because the filtered glitches randomly start reappearing under the influence of process variation. Combining the approaches presented in Chapters 3 and 4, we propose a new statistical MILP formulation, which uses gate sizing, path balancing and dual-threshold techniques to statistically minimize the total power with process variation.

A deterministic MILP using gate sizing, path balancing and dual-V_{th} assignment to reduce the total power consumption is first introduced as a prerequisite and for later modification to consider process variation.

5.1 Deterministic MILP for Total Power Optimization by Dual-V_{th}, Path Balancing and Gate Sizing

In Section 2.3.2, we have discussed two ways of glitch elimination, path balancing and hazard filtering. Path balancing can ensure 100% glitch elimination at a cost of some area and power overhead introduced by the inserted delay elements. It is sensitive to the process variation which will be discussed in Section 5.2.1. Hazard filtering uses gate/transistor sizing to increase inertial delay for glitch filtering. Due to the limitation of the maximum and minimum cell delays that can be achieved in a specific technology or a standard cell library and the unchangeable gate delays on critical paths to meet the performance requirements, hazard filtering cannot guarantee 100% glitch elimination and becomes less efficient for the circuits with many critical paths. But, it has several advantages. Besides its low area and power overheads, this technique applied to ASIC designs is not so sensitive to the process variation, because the discrete gate delays in a standard cell library leave certain relaxed margin for glitch filtering under process variation (see Section 5.2.1). In this section, we combine path balancing and hazard filtering together to get 100% glitch elimination with least power and area overheads.

5.1.1 Gate Sizing for Dynamic Power Reduction

Dynamic power depends on both the circuit switching activity (number of logic transitions) and the power consumption of each logic transition determined mainly by the loading capacitances at gate outputs. Path balancing only eliminates glitches to reduce the number of logic transitions. Through gate sizing, we can further decrease the dynamic power of each transition by the reduction of loading capacitances.

In Chapter 6, to verify the power reduction approaches proposed in Chapters 3 and 4, we implement a simple standard cell library in BPTM 70nm [1] CMOS technology. Each of the 19 standard cells has just one size or driving strength to drive at most five fanouts each with 5fF loading capacitance. This type of cell library makes loading capacitance reduction impossible and allows only a limited optimization space for the dynamic power reduction. The additional power consumed by those inserted delay elements may counteract the eliminated glitch power (see results in Table 6.2).

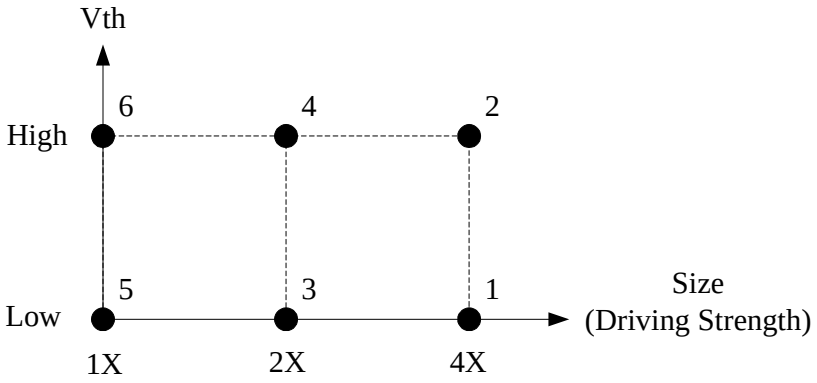


Figure 5.1 Extended cell library with 6 corners for gate sizing.

In a typical ASIC design, each cell in a standard cell library has several sizes to provide different driving strengths. We therefore expand our simple cell library according to Figure 5.1, in which each cell has six corners (1-6) based on its threshold voltage and size. The original library used in Chapters 3, 4 and 6 only has corners 3 (2X, low Vth) and 4 (2X, high Vth). Each cell i has one size ($S_{X2}[i]$), two inertial delay constants ($D_{X2L}[i]$ and $D_{X2H}[i]$) and two leakage current values ($I_{X2L}[i]$ and $I_{X2H}[i]$) for a specified fanout. When expanding this 2-corner library, we let sizes of cell i with 1X and 4X sizes are exactly half and twice $S_{X2}[i]$. Since according to Equation (2.2), subthreshold has an approximately linear relation to the cell width and all standard cells are assumed to have

the minimal lengths to provide the fastest speed, subthreshold leakage of a cell with 1X and 4X sizes are also scaled to half and twice the subthreshold leakage of the same cell with 2X size. Inertial gate delay does not linearly depend upon the gate size, so all inertial delays at six corners are simulated by Spice. After expansion, each cell *i* has three sizes, six inertial delay constants and six leakage current values as shown in Table 5.1.

Table 5.1 Cell characterization in 6-corner cell library

	Corner 1 (4X,LVth)	Corner 2 (4X,HVth)	Corner 3 (2X,LVth)	Corner 4 (2X,HVth)	Corner 5 (1X, LVth)	Corner 6 (1X,HVth)
Size ($Size[i]$)	$2*S_{X2}[i]$	$2*S_{X2}[i]$	$S_{X2}[i]$	$S_{X2}[i]$	$0.5*S_{X2}[i]$	$0.5*S_{X2}[i]$
Delay ($D[i]$)	$D_{X4L}[i]$	$D_{X4H}[i]$	$D_{X2L}[i]$	$D_{X2H}[i]$	$D_{X1L}[i]$	$D_{X1H}[i]$
Subthreshold ($I_{leak}[i]$)	$2*I_{X2L}[i]$	$2*I_{X2H}[i]$	$I_{X2L}[i]$	$I_{X2H}[i]$	$0.5*I_{X2L}[i]$	$0.5*I_{X2H}[i]$

In this 6-corner cell library, cells at corner 1 are the fastest due to their low threshold and largest size devices, although they consume maximum leakage and dynamic power. Cells at corner 6 are most power efficient but have the slowest speed because of their high threshold and smallest size devices. When minimizing the total power consumption in a circuit and providing the highest performance, we use corner 1 as the starting reference case. The optimization of power consumption in circuits implemented by such 6-corner cell library is flexible. To reduce the dynamic logic switching power, we may decrease cell sizes (gate sizing) thereby reducing the loading capacitances. To minimize the leakage, we can either assign high threshold voltages or decrease cell sizes. At the same time, glitches can be eliminated by path balancing and hazard filtering (gate sizing).

5.1.2 Deterministic MILP for Total Power Reduction

This section presents a deterministic MILP formulation to reduce the total power consumption by dual-V_{th} assignment, path balancing and gate sizing.

5.1.2.1 Variables

- Integer [0,1] variables:

  In contrast to the MILP of Chapters 3 and 4, this MILP has 6 integer variables, since each device (functional cell) has 6 alternative choices.

  - *X1L*[*i*], *X2L*[*i*], *X4L*[*i*] - the low threshold version with 1X, 2X and 4X driving strengths respectively, for cell *i*.
  - *X1H*[*i*], *X2H*[*i*], *X4H*[*i*] - the high threshold version with 1X, 2X and 4X driving strengths respectively, for cell *i*.

- Continuous Variables:
  - *Size*[*i*] - size of cell *i.*
  - I_{leak}[*i*] - subthrehold leakage of cell *i.*
  - *D*[*i*] - inertial gate delay of cell *i.*
  - $\Delta d[i,j]$ - delay of a possible delay element that may be inserted at the input of cell *i* on the signal from cell *j*.
  - *T*[*i*] - latest time at which the output of cell *i* can produce an event after the occurrence of an input event at primary inputs of the circuit.
  - *t*[*i*] - earliest time at which the output of cell *i* can produce an event after the occurrence of an input event at primary inputs of the circuit.

5.1.2.2 Constants

- W - weight factors used for tradeoff between leakage and dynamic power
- T_{max} - specified upper bound on critical path delay based on circuit performance requirement.
- $S_{X2}[i]$ - gate size of cell i with 2X driving strength. (In our extended cell library, sizes of cell i with 1X and 4X driving strength are exactly half and twice $S_{X2}[i]$, respectively.)
- $I_{X2L}[i]$, $I_{X2H}[i]$ - subthrehold leakage of cell i with 2X driving strength. (The subthreshold of a cell with 1X or 4X driving strengths is scaled to half or twice the subthreshold leakage of the same cell with 2X driving strength, respectively.)
- $D_{X1L}[i]$, $D_{X2L}[i]$, $D_{X4L}[i]$, $D_{X1H}[i]$, $D_{X2H}[i]$, $D_{X4H}[i]$ - inertial gate delays of cell i at six corners.

5.1.2.3 Objective function

The objective function, minimizing the total power consumption, is given by equation (5.1).

Min {total power consumption}

= Min {leakage power + dynamic power}

= Min {leakage Power +

(logic switching power + dynamic power consumed by the delay elements) }

$$= Min\left\{W \cdot C_1 \sum_i I_{leak}[i] + \left(C_2 \sum_i size[i] + C_3 \sum_i \sum_j \Delta d[i, j]\right)\right\} \quad (5.1)$$

Where C_1, C_2 and C_3 are fitting parameters to let three terms ($C_1\Sigma I_{leak}[i]$, $C_2\Sigma size[i]$ and $C_3\Sigma\Sigma\Delta d[i,j]$) have the same units (μW). To the objective function (3.4) of MILP in Section 3.2.2, we have added $C_2\Sigma size[i]$ to fully utilize the advantage of gate sizing for dynamic power reduction. In a glitch-free circuit optimized by path balancing, dynamic power is composed of two parts, logic switching power and additional dynamic power consumed by the inserted delay elements. Since logic switching power depends on the loading capacitances which are determined by the cell sizes, it can be represented by $C_2\Sigma size[i]$. Extra dynamic power consumed by an inserted delay element also depends on the size of that delay element which is nonlinear related to its delay value. To simply the model, we describe the extra dynamic power introduced by delay elements as $C_3\Sigma\Sigma\Delta d[i,j]$. Therefore, $C_2\Sigma size[i]+C_3\Sigma\Sigma\Delta d[i,j]$ represents the total dynamic power consumption.

To reduce subthreshold leakage (minimize $C_1\Sigma I_{leak}[i]$), we may change the low V_{th} device i to a high V_{th} device, which can approximately reduce 98% of the leakage in BPTM 70nm technology ($V_{dd} = 1V$, Low $V_{th} = 0.20V$, High $V_{th} = 0.32V$). Decreasing size is the other possible way, but with a relatively smaller leakage reduction (i.e., 25% or 50%). Therefore, when minimizing the total subthreshold leakage, the LP solver first tries to do dual-V_{th} assignment, which probably causes the increase of some critical path delays. To remove such timing violations while keeping the total subthreshold as small as possible, some gate delays have to be decreased by gate sizing (enlarging sizes), which increases the dynamic power consumption. Therefore, minimizing $C_1\Sigma I_{leak}[i]$ and minimizing $C_2\Sigma size[i]+C_3\Sigma\Sigma\Delta d[i,j]$ are two conflicting requirements in the objective

function. By adjusting the weight factor *W*, minimal leakage, minimal dynamic power or minimal total power dissipation can be achieved for a specific application.

5.1.2.4 Constraints

- Basic constraints
  - let the LP solver choose one and only one optimal corner for each cell *i*:

$$X1L[i] + X2L[i] + X4L[i] + X1H[i] + X2H[i] + X4H[i] = 1 \tag{5.2}$$

  - leakage of cell *i*:

$$\begin{aligned} I_{leak}[i] = {} & (0.5 \cdot X1L[i] + X2L[i] + 2 \cdot X4L[i]) \cdot I_{X2L}[i] + \\ & (0.5 \cdot X1H[i] + X2H[i] + 2 \cdot X4H[i]) \cdot I_{X2H}[i] \end{aligned} \tag{5.3}$$

  - gate delay of cell *i*:

$$\begin{aligned} D[i] = {} & D_{X1L}[i] \cdot X1L[i] + D_{X2L}[i] \cdot X2L[i] + D_{X4L}[i] \cdot X4L[i] + \\ & D_{X2L}[i] \cdot X1H[i] + D_{X2L}[i] \cdot X2H[i] + D_{X4L}[i] \cdot X4H[i] \end{aligned} \tag{5.4}$$

  - size of cell *i*:

$$Size[i] = \left\{ \begin{matrix} 0.5 \cdot (X1L[i] + X1H[i]) + (X2L[i] + X2H[i]) + \\ 2 \cdot (X4L[i] + X4H[i]) \end{matrix} \right\} \cdot S_{X2}[i] \tag{5.5}$$

- constraints for glitch elimination

For cell *i*, if one of its input is fed by cell *j*'s output, constraint (5.6) makes sure that *T*[*i*] is the latest signal arrival time at the output of cell *i*, and constraint (5.7) ensures that *t*[*i*] is the earliest arrival time.

$$T[i] \geq T[j] + \Delta d[i, j] + D[i] \tag{5.6}$$

$$t[i] \leq t[j] + \Delta d[i, j] + D[i] \tag{5.7}$$

Constraint (5.8) guarantees the output timing window, $T[i]$ - $t[i]$, is always less than the inertial gate delay of cell i, hence glitches will not be generated at cell i's output.

$$D[i] \geq T[i] - t[i] \tag{5.8}$$

- constraint for maximal performance

  To keep the maximal performance, at every primary output k, let,

$$T[k] \leq T_{\max} . \tag{5.9}$$

5.1.3 Results

We compare the dynamic power reduction of C432 (one ISCAS'85 benchmark circuit) implemented by either 2-corner or 6-corner cell library in Table 5.2.

Case 1 (Row 2) - The simple cell library has two corners whose driving strength is 2X. Since all cells implemented by this cell library have only one size, path balancing is the only way to eliminate glitches for dynamic power reduction. Data in column 5, row 2 shows that if the extra dynamic power contributed by the inserted delay elements is ignored, 25.25% power saving can be achieved by path balancing, which means glitches in un-optimized C432 take up 25.3% of total dynamic power consumption. Considering the additional loading capacitances contributed by these inserted delay elements for path balancing, the net dynamic power reduction is only 8.6% (column 7, row 2). Such small dynamic power reduction is almost unacceptable since these inserted delay elements also bring some area overhead except for the dynamic power overhead. If we use gate sizing and path balancing simultaneously, more dynamic power consumption can be reduced.

Table 5.2 Comparison of dynamic power optimization of C432 implemented by 2-corner and 6-corner cell library, respectively.

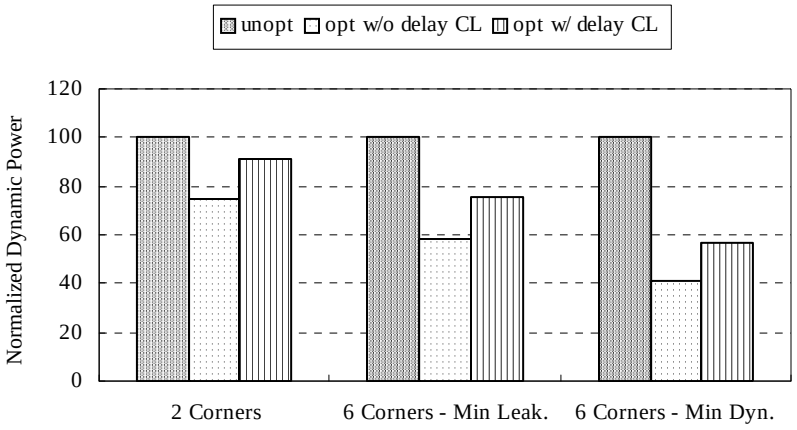
Cell Lib	Weight Factor W	Unopt (μW)	Opt (μW) (w/o delay C_L)	Power Saving	Opt (μW) (w/ delay C_L)	Power Saving	Power of delay elements (μW)
2 corners	-	101	75.5	25.3%	92.3	8.6%	16.8
6 corners	To Min. Leak.	101	58.5	42.1%	76.7	24.1%	18.2
	To Min. Dyn.	101	41.5	58.9%	57.0	43.6%	15.5

Case 2 (Row 3) - The extended cell library has 6 corners whose driving strength can be 1X, 2X or 4X. To make the comparison with case 1 reasonable, we let the unoptimized circuit in case 2 also be implemented by low V_{th} cells with 2X driving strength. When the weight factor (W) is large enough to let MILP's objective function (5.1) emphasize minimizing total subthreshold leakage, 42.1% dynamic power can be reduced by path balancing and gate sizing, or by reducing the number of logic transitions and decreasing loading capacitances simultaneously. This dynamic power reduction decreases to 24.1% when the extra loading capacitances contributed by the inserted delay elements are considered.

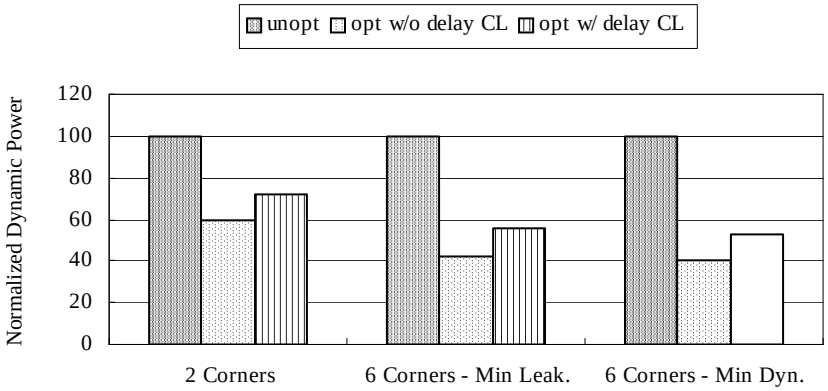
Case 3 (Row 4) – The only difference between case 2 and case 3 is the weight factor used in the MILP formulation. In case 3, the weight factor (W) is small enough to let the CPLEX LP solver [30] emphasize minimizing the total dynamic power, and up to 58.9% of the dynamic power can be saved without considering the additional power introduced by the delay elements. When the extra power contributed by the delay elements is considered, 43.6% of the dynamic power can be saved.

The additional dynamic power contributed by the loading capacitances of the inserted delay elements is shown in column 8. In case 3, although delay elements

consume an additional 15.5μW dynamic power, we still can get more than 40% dynamic power reduction by using path balancing and gate sizing simultaneously, since not only all the glitches are eliminated but also the loading capacitances for some logic transitions are decreased.



(a) C432



(b) C7552

Figure 5.2 Comparison of dynamic power optimization of circuits implemented by 2-corner and 6-corner cell library with different weight factors.

To make the comparison clearer, Figure 5.2 shows the dynamic power reduction in 3 cases for both C432 and C7552. In Chapter 6, Table 6.2 shows that 40.2% of dynamic power in the unoptimized C7552 is consumed by glitches while C432 only has 27.4%, so in both case 2 (6 corners – min leak.) and case 3 (6 corners – min dyn.), C7552 can obtain more dynamic power reduction.

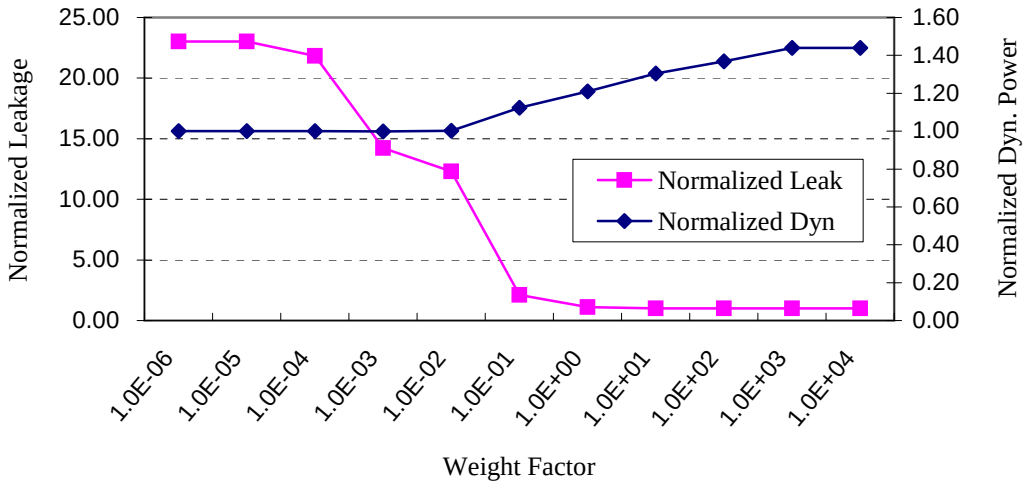


Figure 5.3 Optimization space comparison between leakage and dynamic power of C432 @ 90ºC.

Using the MILP formulation in Section 5.1.2 and carefully adjusting the weight factor *W*, we can get the minimal total power, minimal leakage or dynamic power. In Figure 5.3, when *W* ranges from 10^{-6}, to 10^{4}, normalized leakage is accordingly minimized from its largest value 23.03 (10.36μW) to the minimum value 1 (0.45μW), while normalized dynamic power increases from its smallest value 1 (59.2μW) to the largest value 1.44 (85.2μW). It should be noted that the units for two vertical-axes are different, and the purpose is to show the leakage power has a much larger range of optimization space (23 times) than that of dynamic power (43.9%). This is because

leakage exponentially depends on some process parameters while dynamic power generally has an approximately linear relation to process parameters.

Figure 5.4 shows that the total power consumption changes with the weight factor adjustment. When W equals to 10^{-2}, C432 has the minimum 64.82μW total power consumption if optimized by dual-threshold assignment, path balancing and gate sizing simultaneously. In this case, W is 10^{-2}, which means that the objective function in the MILP model emphasizes dynamic power reduction since, in the optimized C432, leakage is much less than the dynamic power.

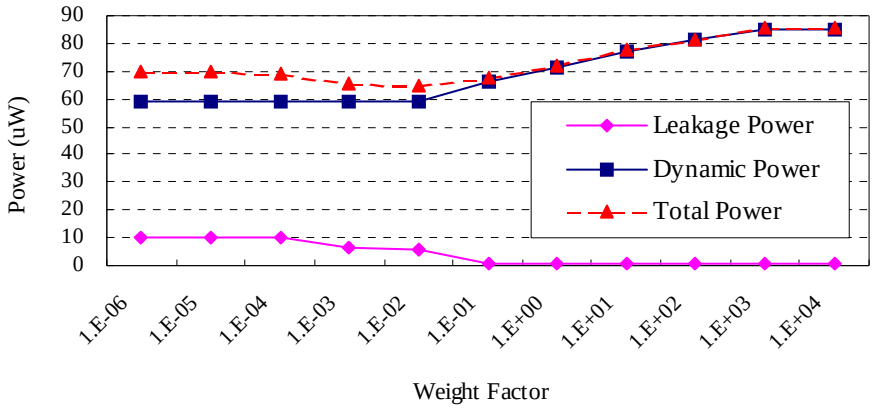


Figure 5.4 Achieving the minimum total power by adjusting the weight factor (W).

Based on this optimal point (0.02, 64.82μW) for the total power optimization without considering any process variation, the statistical MILP formulation can be further adopted to statistically optimize power consumption and consider process variation.

5.2 Statistical MILP for Total Power Optimization

5.2.1 The Impact of Process Variation on Dynamic Power

In Chapter 4, we propose a statistical MILP formulation to minimize the impact of process variation on the subthreshold leakage. Traditionally, due to the approximately linear relation between dynamic power and process parameters, dynamic power is much less sensitive to the process variation. Dynamic power comprises two parts, logic switching power and glitch power, which can be expressed by the following equation:

$$P_{dyn} = \frac{1}{2} C_L V^2 \cdot A \cdot F \;=\; \text{Logic switching power + Glitch power} \qquad (5.10)$$

where A is switching activity and F is the circuit operating frequency.

Logic switching power is directly proportional to the loading capacitances, C_L, which linearly depends upon gate sizes, W (gate width) and L_{eff} (effective gate length). Local (intra-die) process variation causes gate sizes to vary randomly and hence does not affect logic switching power too much. Global (inter-die) process variation changes gate sizes in the same tendency and does vary the logic switching power. However, it does not affect the solution of the MILP formulation, since gate delays and gate sizes in the MILP constraints either increase or decrease with the same percentage when global process variation is considered, and T_{max} is assumed to change accordingly.

The impact of process variation on glitch power is different and more complicated. Glitches are generated if the constraint (5.11) is not satisfied for cell *i*. Since inertial gate delays *D*[i] vary with process variations, inequality (5.11) may change from being satisfied to being violated or vice versa.

$$D[i] \geq T[i] - t[i] \tag{5.11}$$

We consider the impact of global process variation and local process variation on glitch power separately.

- **Impact of global process variation on glitches**

For every gate *i*, its timing window T_i - t_i is actually determined by the two timing paths, the fastest path (*FPath*) and the slowest path (*SPath*) from primary inputs to gate *i*. T_i is the cumulative inertial gate delays along that slowest path, and t_i is the cumulative inertial gate delays along that fastest path, which is shown in equation (5.12).

$$T_i - t_i = \sum_{m \in SPath} d_m - \sum_{n \in FPath} d_n \tag{5.12}$$

Assuming there is $r \cdot 100\%$ (r: 0~1) of global variation applied to the circuit, glitch filtering conditions for gate *i* keep unchanged since both timing window and gate delay vary $r \cdot 100\%$, which are expressed by equations (5.13) and (5.14).

$$T_i' - t_i' = \sum_{m \in SPath} (1+r) d_m - \sum_{n \in FPath} (1+r) d_n = (1+r)\left(\sum_{m \in SPath} d_m - \sum_{n \in FPath} d_n \right) = (1+r)(T_i - t_i) \tag{5.13}$$

$$d_i' = (1+r) d_i \tag{5.14}$$

Therefore, the technique of glitch elimination by path balancing is resistant to the global process variation.

- **Impact of local process variation on glitches**

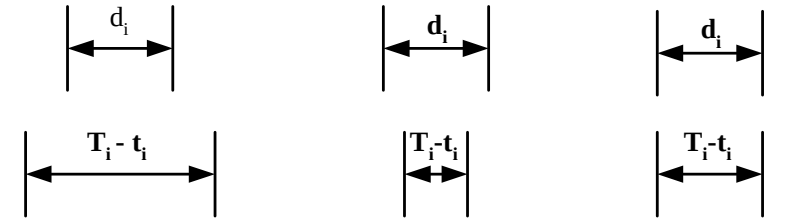
Now, let's consider the impact of local process variation on glitch elimination by path balancing. When local variation is applied to a circuit, as shown in Equation (5.15),

T_i and t_i are the sum of gate delays, which vary randomly, along the slowest and the fastest paths from primary inputs to cell i's inputs, so, T_i - t_i is not very sensitive to the process variations, while d_i does change with the process variation.

$$T_i' - t_i' = \sum_{m \in SPath} (1 + r_m) d_m - \sum_{n \in FPath} (1 + r_n) d_n \tag{5.15}$$

$$d_i' = (1 + r_i) d_i \tag{5.16}$$

As shown in Figure 5.5, there are three possible glitch filtering conditions. Both Figures 5.5(b) and (c) are glitch free while Figure 5.5(a) has a glitch. In an unoptimized (with-glitch) circuit, Figures 5.5(a) or (b) is the much more common condition for one gate, although Figure 5.5(c) is still possible but with the least possibility. On the contrary, in a glitch-free optimized circuit, Figure 5.5(c) is applied to lots of gates because Figure 5.5(a) is always forced to become Figure 5.5(c) by path balancing for glitch elimination.

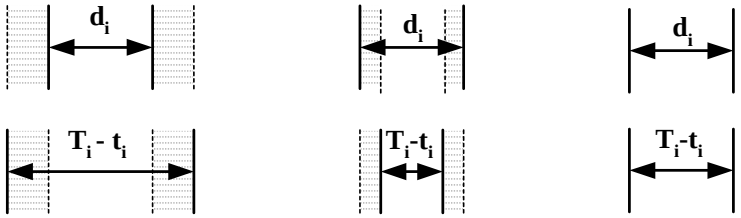


(a) Glitch Not Free (b) Glitch Free (c) Glitch Free

Figure 5.5 Three possible glitch filtering conditions.

With local process variation, Figures 5.6(a) and (b) show that the original condition is not so easily corrupted if only the variation of the timing window or the gate delay falls into the shaded areas, while Figure 5.6(c) is extremely sensitive to the local process

variation, since a slight increase of the timing window or decrease of the gate delay can simply let an original glitch-free gate generate glitches at its output.



(a) Glitch Not Free (b) Glitch Free (c) Glitch Free

Figure 5.6 Three possible glitch filtering conditions under process variation.

This explains why the dynamic power of an unoptimized (with-glitch) circuit is much more resistant to local process variation than that of a glitch-free circuit optimized by path balancing. The glitch-free condition shown in Figure 5.6(c) cannot be really satisfied even with a quite small process variation.

Table 5.3 Normalized dynamic power distribution of un-optimized (with-glitch) C432 under local delay variation.

delay variation	nominal	mean	3*S.D. / mean	(mean-nominal)/nominal
10%	1	0.9995	1.95%	-0.05%
20%	1	0.9987	3.36%	-0.13%
30%	1	0.9978	4.50%	-0.22%

Table 5.3 and Figure 5.7 demonstrate the resistance of unoptimized circuits to the local process variation. We apply 10%, 20% and 30% local delay variations, which are caused by the variation in gate-length-independent $V_{th,}$ to the unoptimized (with-glitch) circuit C432. The largest percentage of the mean value deviated from the nominal value is 0.22% and the maximum spread (3*S.D./mean) is only 4.5%.

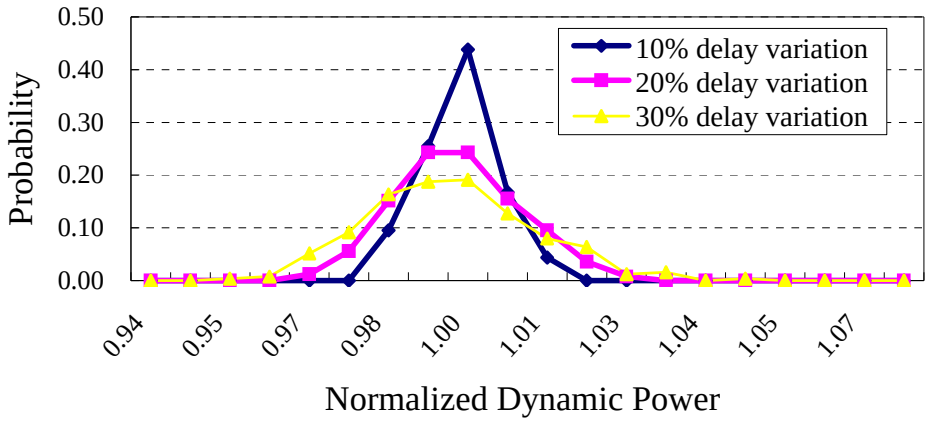


Figure 5.7 Dynamic power distribution of un-optimized (with-glitch) C432 under local delay variation.

The sensitivity of glitch-free circuits optimized by path balancing to the local process variation is illustrated by Table 5.4 and Figure 5.8. Data in Table 5.4 shows that both the mean value and standard deviation of dynamic power distribution increase significantly with the increase of the local process variation. When 30% local variation is applied to the optimized glitch-free C432, its average dynamic power increase 32% and almost equals to the normalized dynamic power (1.34) of unoptimized C432. In Figure 5.8, some samples of optimized C432's dynamic power are even larger than 1.34. It should be mentioned that every sample in Figure 5.8 is larger than the nominal value, 1, which is the expected minimum normalized dynamic power of optimized glitch-free C432 achieved by path balancing. Process variation causes some glitches to be generated in this glitch-free circuit and hence increases the dynamic power.

Table 5.4 Normalized dynamic power distribution of optimized (glitch-free) C432 under local delay variation.

delay variation	nominal	mean	3*S.D. / mean	(mean-nominal)/nominal
10%	1	1.10	5.13%	9.8%
15%	1	1.14	5.34%	13.5%
20%	1	1.23	7.42%	23.2%
30%	1	1.32	9.76%	31.6%

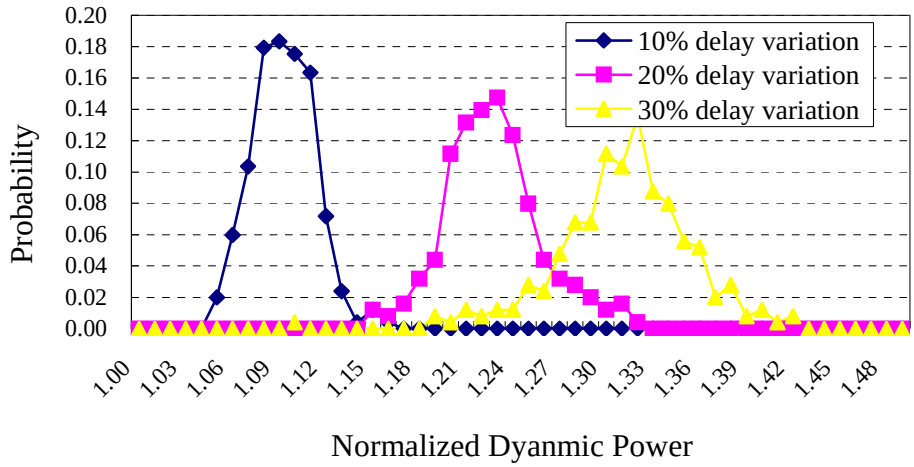


Figure 5.8 Dynamic power distribution of optimized (glitch-free) C432 under local delay variation.

It is remarkable that the advantage of glitch elimination by path balancing is totally lost due to the local process variation. The deterministic approach introduced in Section 5.1.2 is not effective for power optimization with process variation. In the following subsection, we combine the MILP formulation introduced in Chapter 4, and thus a new statistical MILP formulation is proposed to optimize power under process variation and to fully utilize the advantage of path balancing.

5.2.2 Statistical MILP for Power Optimization with Process Variation

Like the statistical MILP formulation presented in Chapter 4, we treat all gate delays and timing window variables as random variables with normal distribution whose standard deviation is σ_r.

5.2.2.1 Variables

- Integer variables:

  Same as the MILP proposed in 5.1.2, this MILP has 6 integer variables since each device has six alternative choices.

  - $X1L[i]$, $X2L[i]$, $X4L[i]$, $X1H[i]$, $X2H[i]$, $X4H[i]$

- Continuous Variables:

  - $\delta[i]$ - relaxed variable for the glitch filtering constraint of cell i. It will be discussed in Section 5.2.2.3.
  - $Size[i]$ - size of cell i.
  - $I_{leak}[i]$ - nominal value of subthrehold leakage of cell i.
  - $u\_D[i]$ - mean of inertial gate delay of cell i.
  - $s\_D[i]$ - standard deviation of inertial gate delay of cell i.
  - $u\_\Delta d[i,j]$ - mean of $\Delta d[i,j]$.
  - $s\_\Delta d[i,j]$ - standard deviation of $\Delta d[i,j]$.
  - $u\_T[i]$ - mean of $T[i]$.
  - $s\_T[i]$ - standard deviation of $T[i]$.
  - $u\_t[i]$ - mean of $t[i]$.
  - $s\_t[i]$ - standard deviation of $t[i]$.

5.2.2.2 Constants

- T_{max} - the maximum expected circuit performance.
- σ_r - standard deviation of the process parameter variations.
- $S_{X2}[i]$ - gate size of cell *i* with 2X driving strength.
- W_1, W_2, W_3 - weight factors.
- $I_{X2L}[i], I_{X2H}[i]$ - nominal values of the subthrehold leakage of cell *i* with 2X driving strength.
- $D_{X1L}[i], D_{X2L}[i], D_{X4L}[i], D_{X1H}[i], D_{X2H}[i], D_{X4H}[i]$ - nominal values of the inertial gate delay of cell *i* at all six corners.

5.2.2.3 Constraints

- Basic constraints
  - Let LP solver choose one and only one optimal corner model for cell *i*.

$$X1L[i] + X2L[i] + X4L[i] + X1H[i] + X2H[i] + X4H[i] = 1$$

  - Nominal value of the subthreshold leakage of cell i:

$$\begin{aligned} u\_I_{leak}[i] = (0.5 \cdot X1L[i] + X2L[i] + 2 \cdot X4L[i]) \cdot I_{X2L}[i] + \\ (0.5 \cdot X1H[i] + X2H[i] + 2 \cdot X4H[i]) \cdot I_{X2H}[i] \end{aligned} \tag{5.17}$$

  - Mean and standard deviation of the gate delay of cell *i*:

$$\begin{aligned} u\_D[i] = D_{X1L}[i] \cdot X1L[i] + D_{X2L}[i] \cdot X2L[i] + D_{X4L}[i] \cdot X4L[i] + \\ D_{X2L}[i] \cdot X1H[i] + D_{X2L}[i] \cdot X2H[i] + D_{X4L}[i] \cdot X4H[i] \end{aligned} \tag{5.18}$$

$$s\_D[i] = \sigma_r \cdot u\_D[i] \tag{5.19}$$

- The size of cell *i*:

$$Size[i] = \begin{Bmatrix} 0.5 \cdot (X1L[i] + X1H[i]) + (X2L[i] + X2H[i]) + \\ 2 \cdot (X4L[i] + X4H[i]) \end{Bmatrix} \cdot S_{X2}[i] \qquad (5.20)$$

- For glitch elimination
  - standard deviation of $\Delta d[i,j]$*:*

$$s\_\Delta d[i, j] = \sigma_r \cdot u\_\Delta d[i, j]) \qquad (5.21)$$

  - auxiliary variables and constraints for calculating the mean and standard deviation of (output) timing window variables.

In the deterministic method, we can use Equations (5.d1-5.d4) to get the output timing window variables, $T[i]$ and $t[i]$, for cell *i*. $T_{in}[i]$ and $t_{in}[i]$ are the variables for the input timing window.

$$T_{in}[i] \geq T[j] + \Delta d[i, j] \qquad (5.d1)$$

$$t_{in}[i] \leq t[j] + \Delta d[i, j] \qquad (5.d2)$$

$$T[i] = T_{in}[i] + D[i] \qquad (5.d2)$$

$$t[i] = t_{in}[i] + D[i] \qquad (5.d4)$$

In statistical approach, Equations (5.d1-5.d4) are expanded as (5.22-5.33). Prefixes *u_* and *s_* represent the mean value and the standard deviation of the corresponding variable, and both $temp\_T_{in}[i]$ and $temp\_t_{in}[i]$ are intermediate variables.

$$u\_T_{in}[i] \geq u\_T[j] + u\_\Delta d[i, j] \qquad (5.22)$$

$$s\_T_{in}[i] = k \cdot (s\_T[j] + s\_\Delta d[i, j]) \qquad (5.23)$$

$$temp\_T_{in}[i] \geq u\_T[j] + u\_\Delta d[i, j] + 3 \cdot s\_T_{in}[i] \qquad (5.24)$$

$$u\_t_{in}[i] \le u\_t[j] + u\_\Delta d[i,j] \quad (5.25)$$

$$s\_t_{in}[i] = k \cdot (s\_t[j] + s\_\Delta d[i,j]) \quad (5.26)$$

$$temp\_t_{in}[i] \le u\_t[j] + u\_\Delta d[i,j] - 3 \cdot s\_t_{in}[i] \quad (5.27)$$

$$s\_T_{in}[i] = (temp\_T_{in}[i] - u\_T_{in}[i])/3 \quad (5.28)$$

$$s\_t_{in}[i] = (u\_t_{in}[i] - temp\_t_{in}[i])/3 \quad (5.29)$$

- mean and standard deviation of timing window variables:

$$u\_T[i] = u\_T_{in}[i] + u\_D[i] \quad (5.30)$$

$$s\_T[i] = k \cdot (s\_T_{in}[i] + s\_D[i]) \quad (5.31)$$

$$u\_t[i] = u\_t_{in}[i] + u\_D[i] \quad (5.32)$$

$$s\_t[i] = k \cdot (s\_t_{in}[i] + s\_D[i]) \quad (5.33)$$

- Glitch filtering constraint in the statistical method:

$$u\_D[i] - 3 \times s\_D[i] \ge (u\_T[i] + 3 \times s\_T[i]) - (u\_t[i] - 3 \times s\_t[i]) \quad (5.34)$$

This constraint can leave certain margin for process variation in advance as shown in Figure 5.6(b) instead of Figure 5.6(c). However, the above worst case constraint is usually too tight to make CPLEX LP [30] solver find a feasible solution. So, we add a relaxed variable ***δ*[i]** to each glitch filtering constraint (5.34).

$$\delta[i] + (u\_D[i] - 3 \times s\_D[i]) \ge (u\_T[i] + 3 \times s\_T[i]) - (u\_t[i] - 3 \times s\_t[i]) \quad (5.35)$$

In the objective function, by minimizing $\Sigma\delta[i]$, CPLEX LP solver will try to find one optimal solution to make as large number of constraints (5.35) satisfied as possible with a

zero $\delta[i]$, which means the glitches of corresponding cells can be truly eliminated even in the worst case condition of process variation. Those constraints only being satisfied with the help of a positive $\delta[i]$ quite likely fail to filter glitches.

- For maximal performance

To keep the maximal performance, at every primary output k, let,

$$u\_T[k] + 3 \times s\_T[k] \le T_{\max} .$$

5.2.2.4 Objective function

The objective function minimizes the impact of process variation on the total power consumption.

Min {the impact of process variation on the total optimal power consumption}

= Min {mean and standard deviation of leakage power +

mean and standard deviation of dynamic power}

$$= Min\left\{ W_1 \cdot C_1 \sum_i I_{leak}[i] + W_2 \cdot \left(C_2 \sum_i size[i] + C_3 \sum_i \sum_j \Delta d[i, j] \right) + W_3 \cdot \sum_i \delta[i] \right\} \quad (5.36)$$

C_1, C_2 and C_3 are fitting parameters to let three terms ($C_1\Sigma I_{leak}[i]$, $C_2\Sigma size[i]$ and $C_3\Sigma\Sigma\Delta d[i,j]$) have the same units (μW). When we talk about process variation, its impact on the mean value and standard deviation of the power consumption should both be considered. For leakage, a smaller mean value automatically means a narrower spread of leakage power distribution since more gates are assigned high V_{th}. Min($C_1\Sigma I_{leak}[i]$) should be enough to minimize the impact of process variation on the total subthreshold leakage. For the dynamic power, standard deviation of the dynamic power distribution is

determined by $\Sigma\delta[i]$ and ($C_2\Sigma size[i]+C_3\Sigma\Sigma\Delta d[i,j]$) affects the average dynamic power. Therefore we should minimize ($C_2\Sigma size[i]+C_3\Sigma\Sigma\Delta d[i,j]$) and $\Sigma\delta[i]$, simultaneously.

The objective function (5.36) is composed of three parts (three single objectives), including, to minimize average leakage power, to minimize average dynamic power and to minimize the standard deviation of the dynamic power. It is actually a multi-objective function and each single objective conflicts with others, for instance, to minimize $\Sigma\delta[i]$ causes the increase of $\Sigma\Sigma\Delta d[i,j]$, and to optimize $\Sigma I_{leak}[i]$ leads to a larger $\Sigma size[i]$, *etc*. It is not easy to get one optimum value for every single objective. What we can do instinctively is to carefully select weight factors, W_1, W_2 and W_3 to make a tradeoff among these three objectives.

It should be noticed that the solution provided by the deterministic MILP in Section 5.1.2 gives us not only a rough image of which one is the dominant power component between leakage and dynamic power but also their exact optimal values (power consumption) in the optimized circuit. Based on that information, we can choose weight factors and add some constraints of the largest allowable minimal leakage or dynamic power in the statistical MILP formulation empirically.

5.2.3 Minimizing Impact of Process Variation on Leakage or Glitch Power

The choice of minimizing the impact of process variation on the leakage or reducing the effect of process variation on the dynamic power is determined by which one is the dominant one between the leakage and the dynamic power, and the circuit applications as well. In a circuit optimized by the deterministic MILP proposed in Section 5.1.2,

- Case 1 - if the optimal leakage is much less than the optimal dynamic power and its large spread due to process variation (for example, 5X difference under 30% global process variation according to Table 4.2) still can be ignored, we need to put much more emphasis on dynamic power resistance to process variation;
- Case 2 - if the optimal leakage is comparable to the optimal dynamic power, and most of the time the circuit in the standby mode, for example, circuits of cell phones, the impact of process variation on the optimal leakage should be minimized with priority definitely since leakage is much more sensitive to the process variation;
- Case 3 - if the optimal leakage is comparable to the optimal dynamic power, and most of the time the circuit is in the active mode, for example, circuits of portable GPS or portable game machines, *etc*., both the mean and standard deviation of the dynamic power distribution should be optimized in the first place.

5.2.3.1 Minimizing the impact of process variation on glitch power

In case 1 and case 3, dynamic power is the dominant component of the total power consumption. Its standard deviation is determined by the number of glitch filtering constraints (5.35) whose $\delta[i]$ are positive values. So, in the MILP objective function (5.37), we first let *W3* be infinitely large to put the highest priority on minimizing $\Sigma\delta$[i].

$$Min\left\{W1\cdot\sum_i I_{leak}[i]+W2\cdot\left(\sum_i size[i]+\sum_i\sum_j \Delta d[i,j]\right)+\underset{W3->\infty}{W3}\sum_i \delta[i]\right\} \tag{5.37}$$

Although MILP tries to minimize $\Sigma\delta[i]$. $\delta[i]$ for some gate may still be positive since the constraint (5.34) is too tight to be satisfied without the help of a positive $\delta[i]$. Every

positive $\delta[i]$ possibly causes the glitch generation at gate i's output. From Table 5.4, we can also see that the average dynamic power linearly increase with the process variation approximately. This increase is contributed by the glitch power which generates under process variation condition. To counteract the increase in the average dynamic power due to those glitches, or to let the really average dynamic power in process variation condition still be close to that one achieved by the deterministic MILP formulation, we have to sacrifice some leakage power to get a smaller logic switching power in advance. This can be achieved by letting *W1* and *W2* both equal to 1 in the MILP objective function (5.38) and adding a new constraint (5.39) to the statistical MILP formation.

$$Min\left\{C_1\sum_i I_{leak}[i]+\left(C_2\sum_i size[i]+C_3\sum_i\sum_j \Delta d[i,j]\right)+\underset{W->\infty}{W3}\sum_i \delta[i]\right\} \tag{5.38}$$

$$C_2\sum_i size[i]+C_3\sum_i\sum_j \Delta d[i,j] < (P_{dyn\_opt}/\rho) \qquad (\rho>1) \tag{5.39}$$

P_{dyn_opt} is the optimal dynamic power obtained by the deterministic MILP in Section 5.1.2 and ρ is a constant determined by the process variation. By letting ρ larger than 1, the statistical MILP formulation can give an optimal circuit which has less dynamic power.

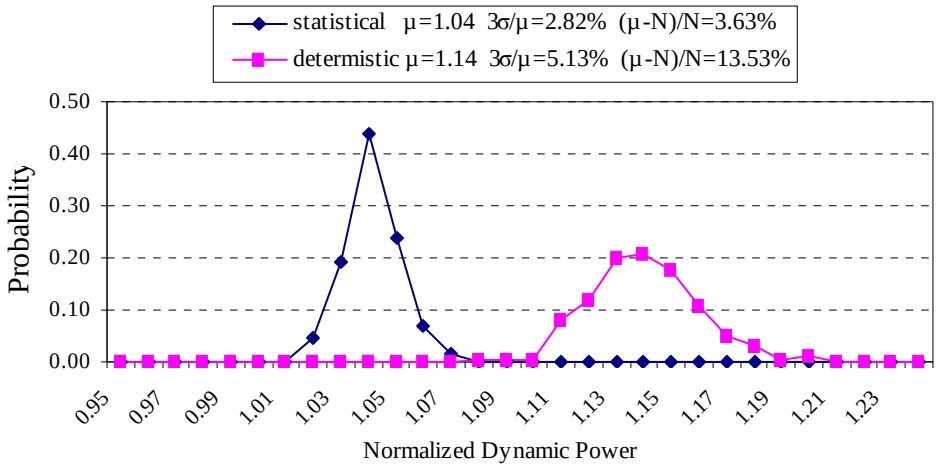


Figure 5.9 Comparison of the impacts of 15% local process variation on the dynamic power in C432 which is optimized by the statistical MILP with the emphasis on the resistance of dynamic power to process variation in Section 5.2.3.1, or by the deterministic MILP in Section 5.1.2. (N=1, is the expected normalized minimum dynamic power in the optimized glitch-free C432).

In C432 optimized by the deterministic MILP formulation in Section 5.1.2, the optimized total power comprises 59.3μW dynamic power and 5.5μW leakage power as shown in Figure 5.4. The data in Table 5.4 shows that with 15% local process variation, its average dynamic power increase 13.53% and with 5.34% standard deviation. To reduce the impact of process variation on its dynamic power, the objective function (5.38) and constraint (5.39) (let P_{dyn_opt}=59.3μW and ρ=1.10) are adopted in the statistical MILP formulation. The two curves in Figure 5.9 show that the average dynamic power only increases 3.63% instead of 13.53%, and standard deviation is also reduced to 2.82% from 5.13% when 15% local process variation is applied to the optimized glitch-free C432, although at a cost of 94% average leakage power increase (from 1.0 to 1.94) and a little bit wider spread of leakage power distribution, which is shown in Figure 5.10.

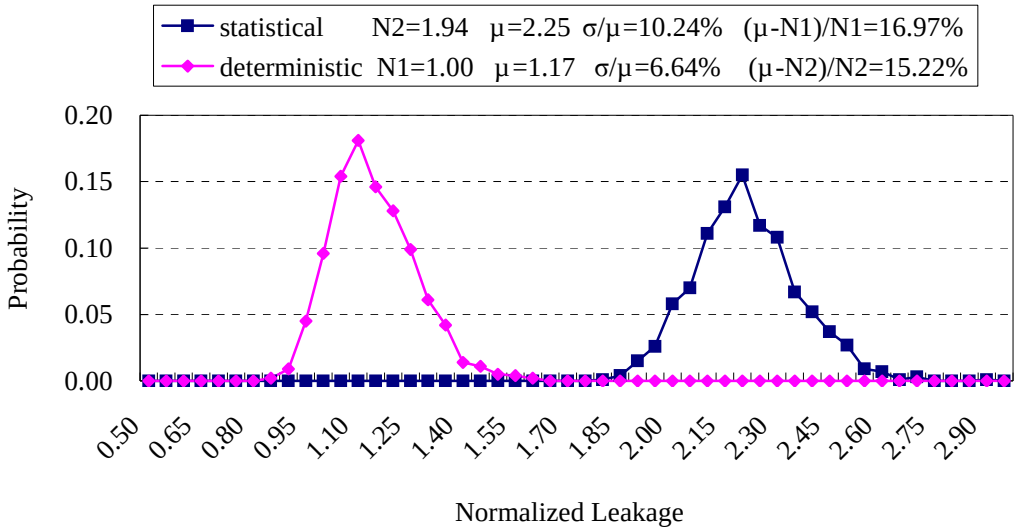


Figure 5.10 Comparison of the impacts of 15% local L_{eff} process variation on the leakage power in C432 which are optimized by the statistical MILP with the emphasis on the resistance of dynamic power to process variation in Section 5.2.3.1, or by the deterministic MILP in Section 5.1.2. (N1 and N2 are the normalized nominal leakage power in the optimized glitch-free C432).

5.2.3.2 Minimizing the impact of process variation on leakage

In case 2, leakage almost equals to or is even larger than the dynamic power. Since leakage is so sensitive to the process variation that we cannot minimize the effect of process variation on the dynamic power by sacrificing leakage any more. The technique of using path balancing to eliminate glitches has to be discarded since the increase in the average dynamic power under process variation may be close to or even larger than the glitch power eliminated by path balancing. To let the leakage of optimized circuits resistant to the process variation, we can still use the MILP proposed in Chapter 4 except every gate has six possible choices instead of just two choices.

5.3 Summary

This chapter first introduces the technique of using gate sizing to reduce dynamic power. Then a deterministic MILP formulation is proposed to optimize the total power consumption by dual-V_{th} assignment, path balancing and gate sizing without considering any process variation. The impact of process variation on dynamic power is analyzed and a statistical MILP formulation is presented to minimize the impact of process variation on the dynamic power by giving up some leakage power if the dynamic power is still the dominant one under process variation. Figure 5.11 gives the flowchart of how to make a decision as to which one, leakage or dynamic power, should be optimized considering process variation.

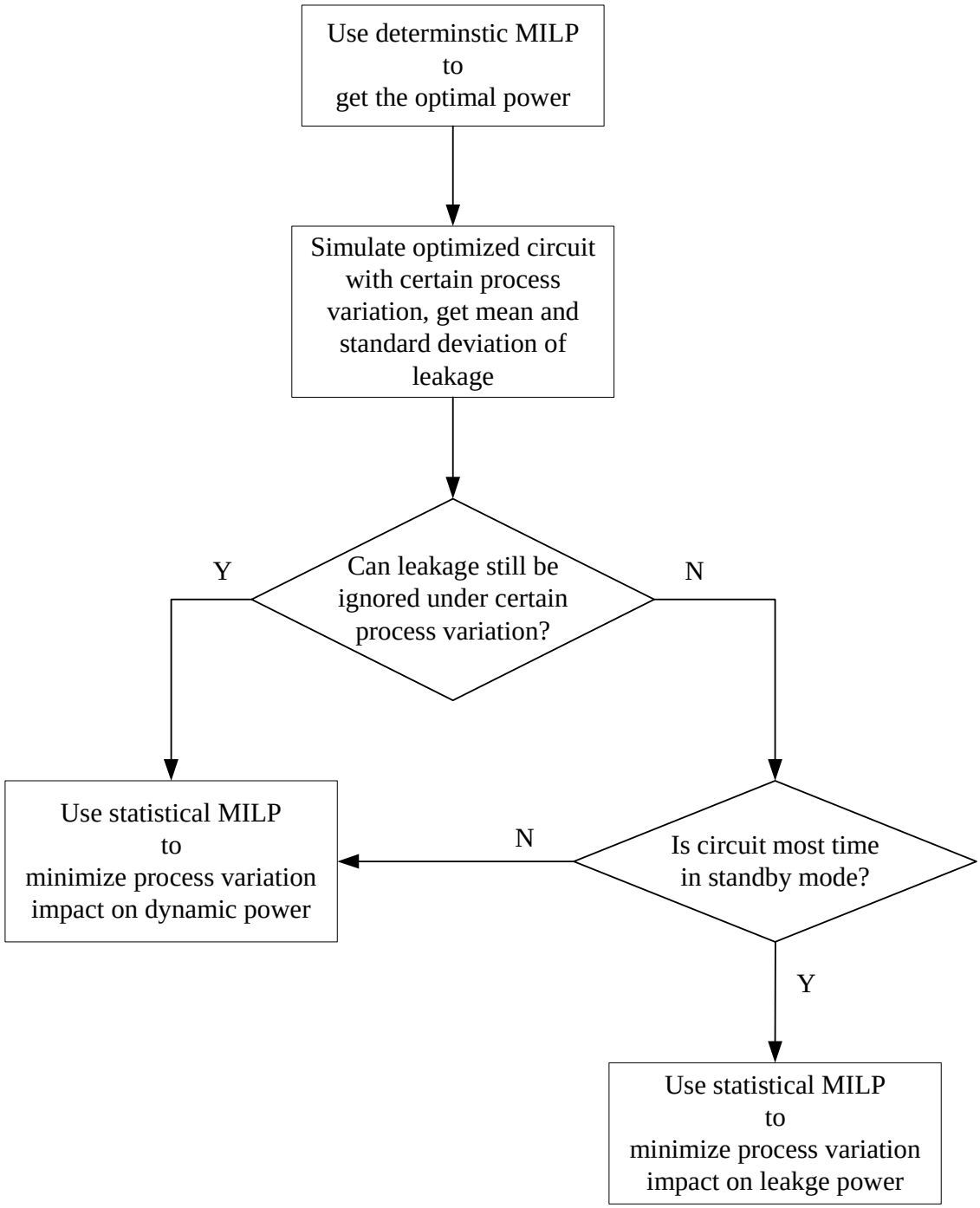


Figure 5.11 Flowchart of making a decision as to which one, leakage or dynamic power, should be optimized with process variation.

CHAPTER 6 RESULTS

To study the increasingly dominant effect of leakage power, we use the BPTM 70nm CMOS technology [1]. Low V_{th} for NMOS and PMOS devices are 0.20V and – 0.22V, respectively. High V_{th} for NMOS and PMOS are 0.32V and – 0.34V, respectively. We regenerated the netlists of ISCAS'85 benchmark circuits using a 2-corner cell library in which the maximum gate fanin is 5. Two look-up tables for gate delays and leakage currents, respectively, of each type of cell were constructed using Spice simulation. A C program parses the netlist and generates the constraint set for the CPLEX LP solver in the AMPL software package [30]. CPLEX then gives the optimal V_{th} assignment as well as the value and position of every delay element. The dynamic power is estimated by an event driven logic simulator that incorporates an inertial delay glitch filtering analysis.

6.1 Results of Deterministic MILP (Chapter 3) for Total Power Optimization

6.1.1 Leakage Power Reduction

The results of leakage power reduction for ISCAS'85 benchmark circuits are shown in Table 6.1. Here the objective of the MILP in Section 3.2 was set to minimize the leakage alone. All $\Delta d_{i,j}$ variables were forced to be 0 and constraints (3.9) and (3.10)

were suppressed. The numbers of gates in column 2 are for our gate library and differ from those in the original benchmark netlists. T_c in column 3 is the minimum delay of the critical path when all gates have low V_{th}. This was determined by the LP discussed in Section 3.2 in the paragraph following Equation (3.16). Column 4 shows the total leakage current with all gates assigned low V_{th}. Column 5 shows the optimized circuit leakage current with gate V_{th} reassigned according to the MILP optimization. Column 6 shows the leakage reduction (%) for optimization without sacrificing any performance. Column 9 shows the leakage reduction with 25% performance sacrifice.

Table 6.1 Leakage reduction alone due to dual-V_{th} assignment (27°C).

Circuit name	# gates	T_c (ns)	Unopt I_{leak} (μA)	Optimized (T_{max}= T_c)			Optimized (T_{max}= $1.25T_c$)		
				I_{leak} (μA)	Leakage reduction	Sun OS 5.7 CPU s	I_{leak} (μA)	Leakage reduction	Sun OS5 .7 CPU s
C432	160	0.751	2.620	1.022	61.0%	0.42	0.132	95.0%	0.3
C499	182	0.391	4.293	3.464	19.3%	0.08	0.225	94.8%	1.8
C880	328	0.672	4.406	0.524	88.1%	0.24	0.153	96.5%	0.3
C1355	214	0.403	4.388	3.290	25.0%	0.1	0.294	93.3%	2.1
C1908	319	0.573	6.023	2.023	66.4%	59	0.204	96.6%	1.3
C2670	362	1.263	5.925	0.659	90.4%	0.38	0.125	97.9%	0.16
C3540	1097	1.748	15.622	0.972	93.8%	3.9	0.319	98.0%	0.74
C5315	1165	1.589	19.332	2.505	87.1%	140	0.395	98.0%	0.71
C6288	1177	2.177	23.142	6.075	73.8%	277	0.678	97.1%	7.48
C7552	1046	1.915	22.043	0.872	96.0%	1.1	0.445	98.0%	0.58

From Table 6.1, we see that by V_{th} reassignment, the leakage current of most benchmark circuits is reduced by more than 60% without any performance sacrifice (column 6). For several large benchmarks leakage is reduced by 90% due to a smaller percentage of gates being on critical paths. However, for some highly symmetrical

circuits, which have many critical paths, such as C499 and C1355, the leakage reduction is less. Column 9 shows that the leakage reduction reaches the highest level, around 98%, with some performance sacrifice.

The curves in Figure 6.1 show the relation between normalized leakage power and normalized critical path delay in a dual-V_{th} process. Unoptimized circuits with all low V_{th} gates are at point (1, 1) and have the largest leakage power and smallest delay. With optimal V_{th} assignment, leakage power can be reduced sharply by 61% (from point (1, 1) to point (1, 0.4)) for C432 or 88% (from point (1, 1) to point (1, 0.1)) for C880, depending on the circuit, without sacrificing any performance. When normalized T_{max} becomes greater than 1, i.e., we sacrifice some performance, leakage power further decreases with a slower decreasing trend. When the delay increase is more than 30%, the leakage reduction saturates at about 98%. Thus, Figure 6.1 provides a guide for making tradeoffs between leakage power and performance.

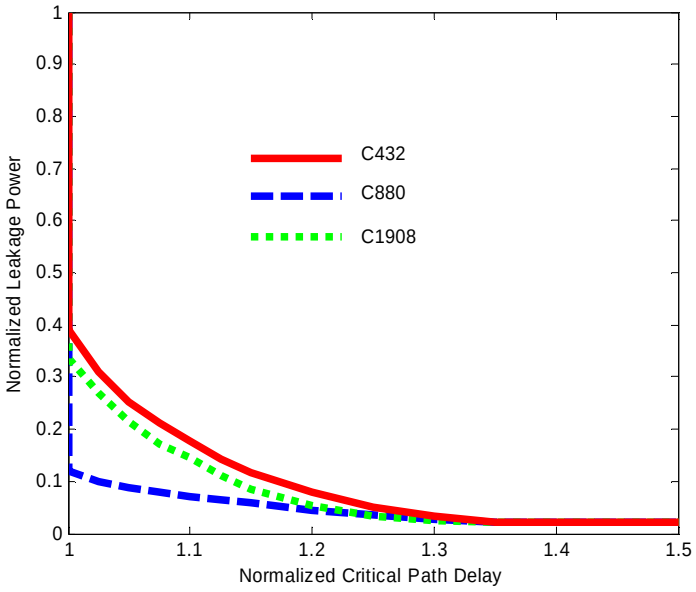


Figure 6.1 Tradeoffs between leakage power and performance.

6.1.2 Leakage, Dynamic Glitch and Total Power Reduction

The leakage current increases with temperature because V_T (thermal voltage, kT/q) and V_{th} both depend on the temperature. Our Spice simulation shows that for a 2-input NAND gate with low V_{th}, when temperature increases from 27°C to 90°C, the leakage current increases by a factor of 10. For a 2-input NAND gate with high V_{th}, this factor is 20.

The leakage in our look-up table is from simulation for 27°C operation. To manifest the dominant effect of the leakage power, we estimate the leakage currents at 90°C by multiplying the total leakage current obtained from CPLEX LP solver [30] by a factor between 10 and 20 as determined by the proportion of low to high threshold transistors.

The dynamic power is estimated by a glitch filtering event driven simulator, and is given by

$$P_{dyn} = \frac{E_{dyn}}{T} = \frac{0.5 \cdot C_{inv} \cdot V_{dd}^{\ 2} \cdot \sum_i T_i FO_i}{1000(1.2 \cdot T_c)} \tag{6.1}$$

where C_{inv} is the gate capacitance of an inverter, T_i is the number of transitions at the output of gate i when 1,000 random vectors are applied at PIs, and FO_i is the number of fanouts for gate i. The vector period is assumed to be 20% greater than the critical path delay, T_c. By simulating each gate's number of transitions, we can estimate the glitch power reduction.

When path balancing is used to eliminate glitches, the additional loading capacitances contributed by the inserted delay elements consume extra dynamic power. Whether the technique of path balancing is effective depends on the ratio of this dynamic

power overhead to the eliminated glitch power. Data in column 3 of Table 6.2 show that less than 10% dynamic power reduction can be achieved for some circuits, for instance, C432, C1908 and C2670, when the loading capacitances of the delay elements are considered. This is mainly because we use a 2-corner cell library which has a limited optimization space. As we discussed and illustrated in Section 5.1, using a 6-corner cell library, normally we can achieve more dynamic power reduction since this type of cell library makes it possible to eliminate glitches by path balancing and to reduce loading capacitances for each logic transition by gate sizing simultaneously.

Table 6.2 Comparison of the percentage of glitches in unoptimized circuits with the real percentage of dynamic power reduction achieved by path balancing considering the additional loading capacitances contributed by the delay elements.

Cirt. Name	Glitch % in Un-opt Circuits	Dynamic Power reduction W/ C_L of delay elements
C432	27.4 %	8.63 %
C499	29.0 %	18.13%
C880	27.8 %	16.23%
C1355	43.5 %	35.79%
C1908	22.4 %	8.39%
C2670	21.6 %	7.42%
C3540	31.5 %	14.04%
C5315	34.6 %	12.08%
C6288	76.0 %	68.73%
C7552	40.2 %	27.74%

To demonstrate the projected dominant effect of leakage power in a sub-micron CMOS technology, we compare the leakage power and dynamic power at 90°C in Table 6.3. "All low V_{th}" means the unoptimized circuit that has all low threshold gates, and "Dual V_{th}" means the optimized circuit whose V_{th} has been optimally assigned for

minimum leakage. Column 6 gives the dynamic power of the optimized design, which is further reduced as shown in column 7 when glitches are eliminated by path balancing and the power overhead contributed by the delay elements is considered. We observe that for 70nm BPTM CMOS technology at 90°C, unoptimized leakage power (column 3) of some large ISCAS'85 benchmark circuits can account for about one half or more of the total power consumption (column 9). With V_{th} reassignment, the optimized leakage power of most benchmark circuits is reduced to around 10%. With further glitch (dynamic) power reduction, the average total power reduction for ISCAS'85 benchmark is 40%. Some have a total reduction of up to 70%.

Table 6.3 Leakage, glitch and total power reduction for ISCAS'85 benchmark circuits (90°C).

Cirt. Name	# gates	Leakage Power (μW)			Dynamic Power (μW)			Total Power (leakage+dynamic) (μW)		
		All low V_{th}	Dual V_{th}	Reduc. %	Dual V_{th}	Delay Opt.	Reduc. %	All low V_{th}	Dual V_{th} + Del Opt.	Reduc %
C432	160	35.77	11.87	66.8%	101.0	73.3	8.63 %	136.8	104.15	23.86%
C499	182	50.36	39.94	20.7%	225.7	160.3	18.13%	276.1	224.72	18.61%
C880	328	85.21	11.05	87.0%	177.3	128.0	16.23%	262.5	159.57	39.21%
C1355	214	54.12	39.96	26.3%	293.3	165.7	35.79%	347.4	228.29	34.29%
C1908	319	92.17	29.69	67.8%	254.9	197.7	8.39%	347.1	263.20	24.17%
C2670	362	115.4	11.32	90.2%	128.6	100.8	7.42%	244.0	130.38	46.57%
C3540	1097	302.8	17.98	94.1%	333.2	228.1	14.04%	636.0	304.40	52.14%
C5315	1165	421.1	49.79	88.2%	465.5	304.3	12.08%	886.6	459.06	48.22%
C6288	1177	388.5	97.17	75.0%	1691	405.6	68.73%	2079.7	625.95	69.90%
C7552	1046	444.4	18.75	95.8%	380.9	227.8	27.74%	825.3	293.99	64.38%

6.1.3 Tradeoff Between Glitch Power Reduction and Area/Power Overhead Contributed by the Delay Elements

The area overhead due to the inserted delay elements is somewhat large. From Table 6.4, we observe that the number of delay elements (Δdi #) is almost equal to the number of gates (Gates #), except for C1355. If we assume that the average number of transistors in a gate is 4 (e.g., consider a 2-input NAND gate), and each delay element implemented by a CMOS transmission gate has 2 transistors, the rough area overhead will be around 50% due to delay element insertion. The main reason is that our cell library has some complex gates, for example, AOI (AND-OR-INVERT) gates whose fanin number may be as large as 5. Some NAND or NOR gates can also have as large as 4 inputs. As a result, it is very possible that more than one delay buffer is inserted for a gate. The solution is to use a simpler and smaller cell library which will be used in our following research.

Table 6.4 Number of delay elements for optimization.

Circuit	Gates #	Δdi #
C432	160	160
C499	182	128
C880	328	303
C1355	214	112
C1908	319	313
C2670	362	330
C3540	1097	1258
C5315	1165	1198
C6288	1177	1307
C7552	1046	845

Considering the usually large routing area in an ASIC chip, and the fact that a large percentage of delay elements have quite small delays (see the following discussion in this section) and hence small sizes, the actual area overhead should be much less than 50%.

We also applied the path balancing technique to an ADI (Analog Devices Inc.) RFID chip which is implemented in TSMC 0.35um CMOS technology and has 46,000 placeable cells (39,000 combinational cells and 7,000 sequential cells). The power simulation results by PrimePower [5] show that 11.8% of the logic transitions are glitches which consume 8% of the dynamic power. Here the internal logic switchings inside of a standard cell are not considered. Although this RFID chip does not consume too much glitch power, the analysis of the values and number of the delay elements is still instructive.

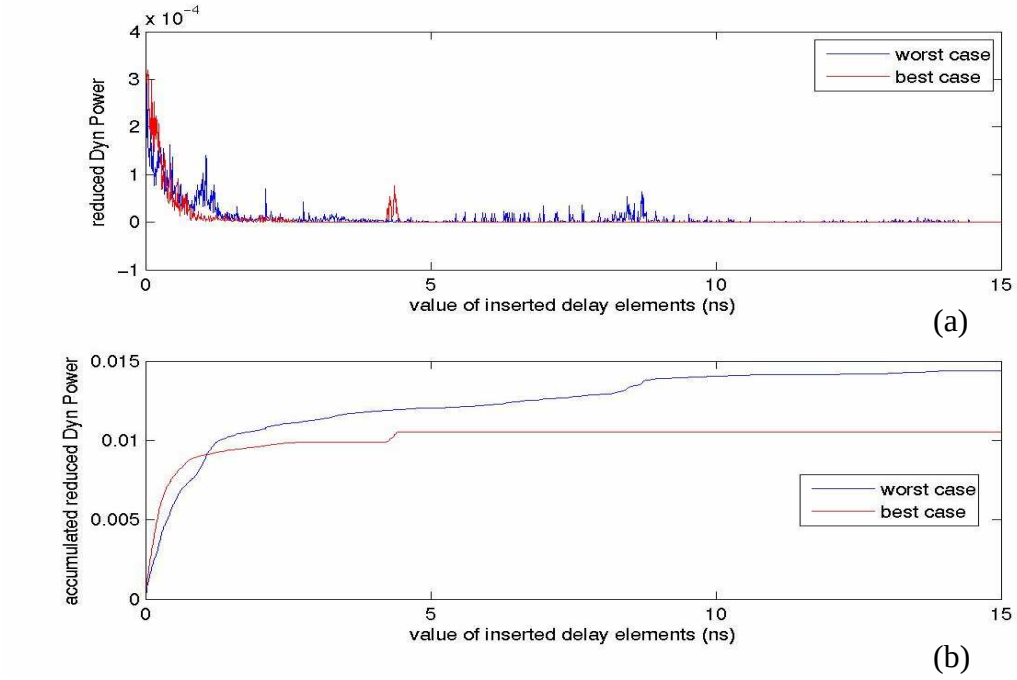


Figure 6.2 (a) dynamic power reduction by delay elements with a certain delay *D*, and (b) cumulative dynamic power reduction by delay elements with delay 0~D.

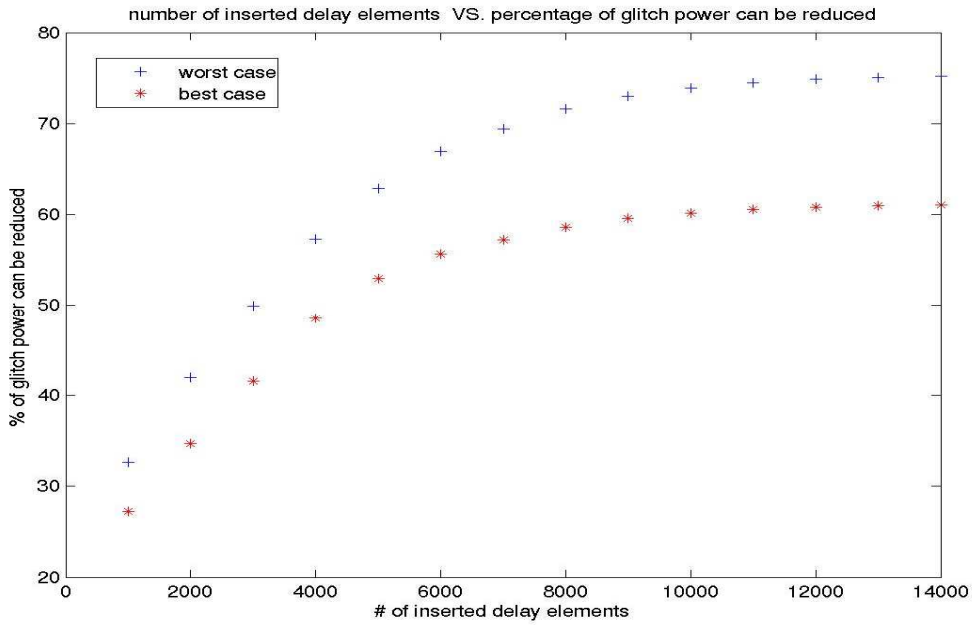


Figure 6.3 The relation between the number of inserted delay elements (sorted by their contribution to the dynamic power reduction) and the corresponding percentage of glitch power reduction

Figure 6.2(a) gives the PDF (probability distribution function) of the delay elements, or dynamic power reduction by delay elements with a certain delay *D*. It shows that most of the delay elements inserted for glitch elimination have small delays. This coincides with the nature of the circuit structure in a high speed ASIC design. The logic depth of any combinational logic between two flip-flops cannot be very large in a high speed ASIC chip and hence the timing window determining the value of a delay element is not wide. Figure 6.2(b) gives the CDF (cumulative distribution function) of the delay elements, or the cumulative dynamic power reduction by delay elements with delay 0~*D*. It is found that delay elements whose delays are larger than 5ns or 10ns for the best case or worst case, respectively, contribute very little to the dynamic power reduction.

Therefore, Figure 6.2 gives us guidance for the selection of the delay elements when a standard cell library of delay elements is constructed.

The relation between the number of inserted delay elements and the corresponding percentage of glitch power reduction is shown in Figure 6.3. Delay elements are assorted by their contribution to the dynamic power reduction. The fist 10,000 delay elements play a much more important role in glitch elimination, while the remaining 4,000 cells' contribution is very small. Figure 6.3 actually provides circuit designers a clue of how to make a tradeoff between glitch reduction and power/area overhead introduced by those delay elements. It should be noted that the glitches propagated at the outputs of buffers and inverters disappear automatically when all the paths are balanced. In this RFID chip, this type of glitches consumes 25% and 39% of the total glitch power for the worst case and best case respectively. Therefore, the maximum glitch power contributed by all the remaining glitches is 75% and 61% of the total glitch power for the worst case and best case respectively.

6.2 Results of Statistical MILP (Chapter 4) for Leakage Optimization

To compare the power optimization results of the statistical MILP with those from the deterministic approach, we assume that all the gates have the same c_{i1} and c_{i2} (sensitivities of gate delay to the variation of different process parameters) in equation (4.9). Therefore, each gate has the same r_i and we assume $3\sigma/\mu$ of r_i is 15%. This assumption is only for the simplicity and does not change the efficacy of the statistical approach.

In the deterministic method, the worst case is applied, which means all gate delays increase 15% and hence T_{max} increases 15% accordingly. To make the comparison between the statistical method and the deterministic approach reasonable, T_{max} in the statistical approach is also 115% of the original value.

Table 6.5 Comparison of leakage power saving due to statistical modeling with two different timing yields (η).

Circuit			*Deterministic Optimization (η = 100%)*		*Statistical Optimization (η = 99%)*			*Statistical Optimization (η = 95%)*		
Name	# gate	Unopt. Leak. Power (μW)	Opt. Leak. Power (μW)	Run Time (s)	Opt. Leak. Power (μW)	Extra Power Saving	Run Time (s)	Opt. Leak. Power (μW)	Extra Power Saving	Run Time (s)
C432	160	2.620	1.003	0.00	0.662	33.9%	0.44	0.589	41.3%	0.32
C499	182	4.293	3.396	0.02	3.396	0.0%	0.22	2.323	31.6%	1.47
C880	328	4.406	0.526	0.02	0.367	30.2%	0.18	0.340	35.4%	0.18
C1355	214	4.388	3.153	0.00	3.044	3.5%	0.17	2.158	31.6%	0.48
C1908	319	6.023	1.179	0.03	1.392	21.7%	11.21	1.169	34.3%	17.5
C2670	362	5.925	0.565	0.03	0.298	47.2%	0.35	0.283	49.8%	0.43
C3540	1097	15.622	0.957	0.13	0.475	50.4%	0.24	0.435	54.5%	1.17
C5315	1165	19.332	2.716	1.88	1.194	56.0%	67.63	0.956	64.8%	19.7
C7552	1046	22.043	0.938	0.44	0.751	20.0%	0.88	0.677	27.9%	0.58
Average of ISCAS'85 benchmarks				0.24		29.2%	9.04		41.3%	4.64
ARM7	15.5k	686.56	495.12	15.69	425.44	14.07%	36.79	425.44	14.07%	36.4

In Table 6.5, columns 4, 6 and 9 give the optimized leakage power by deterministic MILP, by statistical MILP with 99% timing yield and by statistical MILP with 95% timing yield. From Table 6.5, we see that compared to the deterministic method, which uses the fixed values, when we use statistical models for gate delay and subthreshold leakage current, ISCAS85 benchmarks can achieve on average 29% greater leakage power saving with 99% timing yield and 41% greater power saving with 95% timing yield. The reason is that statistical model has a more flexible optimization space,

while the deterministic approach assumes the worst case. For C499 and C1355, which have many critical paths due to their extremely symmetrical circuit structures, the optimization space is limited and therefore the additional power saving contributed by optimization is much smaller, especially with the higher timing yield (99%). It is also obvious that with a decreased timing yield, higher power saving can be achieved due to the relaxed timing constraints, resulting in a larger optimization space.

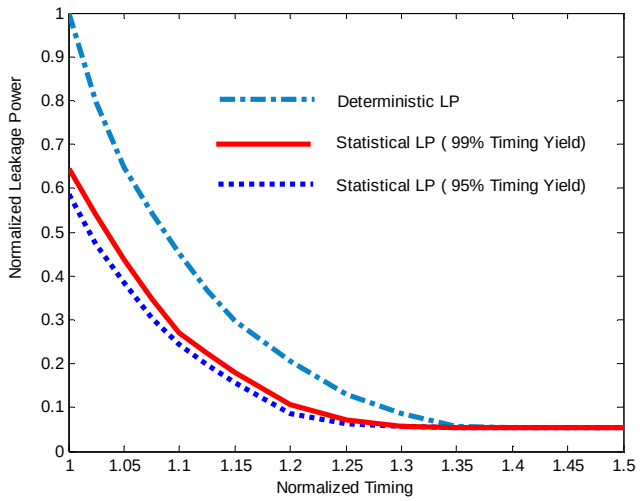


Figure 6.4 Power-delay curves of deterministic and statistical approaches for C432.

Figure 6.4 shows the power-delay curves for C432's leakage optimization by deterministic and statistical approaches. The starting points of the three curves, (1,1), (1,0.66) and (1,0.59), indicate that if we can reduce the leakage power to 1 unit by deterministic approach, 0.65 unit and 0.59 unit leakage power can be achieved by using statistical approach with 99% and 95% timing yields, respectively. The lower the timing yield, the higher the power saving. With a further relaxed T_{max}, all three curves will give more reduction in leakage power because more gates will be assigned high V_{th}.

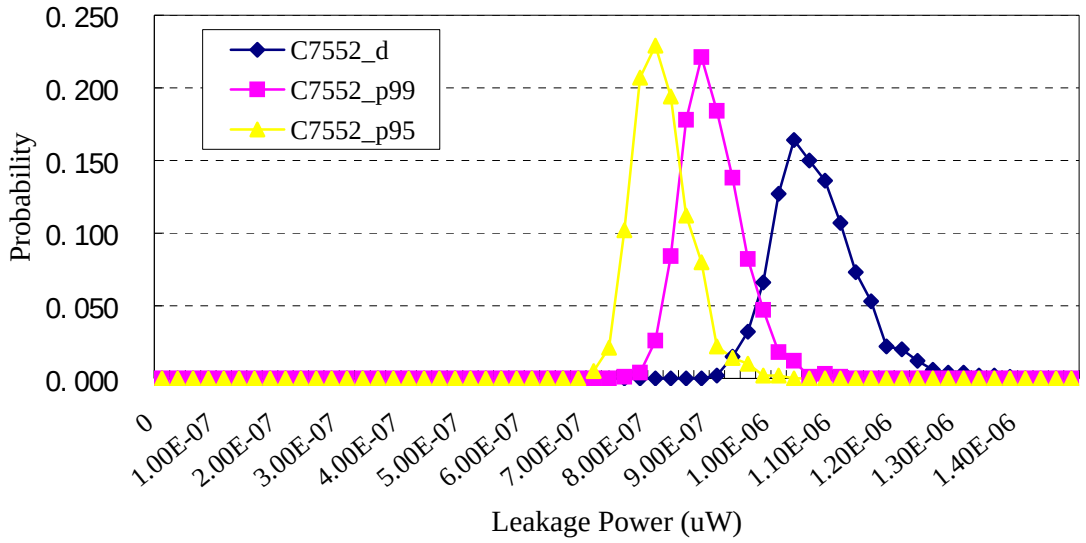


Figure 6.5 Leakage power distribution of dual-V_{th} C7552 optimized by deterministic method, statistical methods with 99% and 95% timing yields, respectively.

Figure 6.5 shows a clear comparison of the leakage power distributions of dual-V_{th} C7552 optimized by the deterministic method, and the statistical methods with 99% and 95% timing yield, respectively. We can see that both mean and standard deviation of C7552's leakage distribution are reduced by statistical approaches as compared to the deterministic method. Although not very obvious, leakage optimization with 95% timing yield indeed has a smaller spread than that with 99% timing yield.

The reason for the narrower leakage distribution and lower average leakage lies in the fact that more high threshold gates can be assigned by the statistical method compared to the deterministic method. Because, when optimizing the leakage and considering process variation by the deterministic approach, we have to analyze the worst case which is too pessimistic. The leakage in high V_{th} gates is less sensitive to the process variation, because although high V_{th} gates may have the same percentage of leakage variation as low V_{th} gates, the absolute variation in high V_{th} gates is certainly much

smaller. Therefore, a higher percentage of high V_{th} gates in a dual-V_{th} circuit ensures a narrower spread and a lower mean of leakage power.

Table 6.6 Monte Carlo Spice simulation results for the mean and the standard deviation of the leakage distributions of ISCAS'85 circuits optimized by deterministic method, statistical methods with 99% and 95% timing yields, respectively.

Circuit		*Deterministic Optimization (η = 100%)*			*Statistical Optimization (η = 99%)*			*Statistical Optimization (η = 95%)*		
Name	# gates	Nom. Leak. (nW)	Mean Leak. (nW)	S.D. (nW)	Nom. Leak. (nW)	Mean Leak. (nW)	S.D. (nW)	Nom. Leak. (nW)	Mean Leak. (nW)	S.D. (nW)
C432	160	0.907	1.059	0.104	0.603	0.709	0.074	0.522	0.614	0.069
C499	182	3.592	4.283	0.255	3.592	4.283	0.255	2.464	2.905	0.197
C880	328	0.551	0.645	0.086	0.430	0.509	0.080	0.415	0.491	0.079
C1355	214	3.198	3.744	0.200	3.090	3.606	0.202	2.199	2.610	0.175
C1908	319	1.803	2.123	0.170	1.356	1.601	0.116	1.140	1.341	0.127
C2670	362	0.635	0.750	0.078	0.405	0.473	0.046	0.395	0.461	0.043
C3540	1097	1.055	1.243	0.119	0.527	0.611	0.032	0.493	0.575	0.031
C5315	1165	2.688	3.128	0.165	1.229	1.420	0.088	1.034	1.188	0.067
C7552	1045	0.924	1.073	0.069	0.774	0.903	0.049	0.701	0.823	0.045
Average of ISCAS'85 benchmarks				0.138			0.105			0.093

In global process variation, all the gate delays have the same percentage of variation, and hence no effect on the timing window constraints in the statistical MILP, which means the assignment of the dual threshold voltages is kept unchanged. On the other hand, subthreshold current is most sensitive to the L_{eff} variation. Therefore, in Table 6.6, we simulate the leakage distributions of all the deterministically and statistically optimized ISCAS'85 benchmark circuits with ***local*** $\boldsymbol{L_{eff}}$ variation ($3\sigma/\mu$=15%) by Spice. Just as expected, almost all of the mean and standard deviations of the leakage distributions are decreased by statistically approaches. Narrower spread and lower mean

can be achieved by the statistical method with 95% timing yield compared to that with 99% timing yield.

6.3 Run Time of MILP Algorithms

The run time of MILP is always a big concern since its complexity is exponential in the number of variables and constraints of the problem in the worst case. However, our experimental results show that the real computing time may depend on the circuit structure, logic depth, *etc*., and may not be exponential.

The CPU times shown in columns 7 and 10 of Table 6.1 are for the deterministic MILP in Chapter 3. From the data in Table 6.1, it is hard to express any relation between the CPU time and the problem size, such as the number of gates in the circuit. For example, MILP solution time for the 1046-gate C7552 is only 1.1 CPU seconds, which is much less than 140 CPU seconds used for the 1165-gate C5315. Even for the same size problems, different constraints require varying solution times. Consider the 1177-gate C6288 circuit as an example. When the timing constraints for primary outputs (POs) are relaxed by 25%, CPU time decreases from 277 CPU seconds to 7.48 CPU seconds. As a result, MILP formulation may still solve some very large size circuits and provide a possibly better solution to dual-V_{th} assignment problem through global optimization.

Running on a 2.4GHz AMD Opteron 150 processor with 3GB memory, many CPU run times for solving the statistical MILP problem (Chapter 4) were less than one second (columns 5, 8 and 11 in Table 6.5). This is an advantage over other techniques [61] because we achieve 30% more leakage reduction with 99% timing yield but in much less CPU time.

Besides ISCAS'85 benchmark circuits, we also optimized the leakage for an ARM7 IP core, which has 15,500 combinational cells and 2,400 sequential cells implemented in TSMC 90nm CMOS process. The experimental results in the last row of Table 6.5 show that 14% more leakage reduction is achieved with 37 seconds run time and partly demonstrate the feasibility of applying our MILP approach to real circuits.

Although today's SOC may have over one million gates, it always has a hierarchical structure. MILP constraints can be generated for submodules at a lower level and the run times will be determined by the number of gates in the individual submodules. Such a technique may not guarantee a global optimization, but still would get a reasonable result within acceptable run time.

6.4 Summary

Experimental results are presented and discussed in this chapter. The results show that the deterministic MILP formulation proposed in Chapter 3 for total power reduction by path balancing and dual-V_{th} assignment can achieve on average 40% total power reduction. If combining with the gate sizing technique discussed in Chapter 5, more power reduction can be obtained. The statistical MILP proposed in Chapter 4, for minimizing the impact of process variation on leakage power, can achieve 30% more leakage power reduction compared to the deterministic MILP formulation. Whether is it necessary to minimize the impact of process variation on dynamic power depends upon the circuit applications and which one is the dominant power component in the optimized circuit, so we only propose the corresponding statistical MILP formulation in Chapter 5 and do not give more detailed results in this chapter.

CHAPTER 7 CONCLUSION AND FUTURE WORK

In this chapter, we summarize the entire work of this dissertation and provide some suggestions for future research.

7.1 Conclusion

With the continuing trend of technology scaling, leakage power has become a main contributor to power consumption. Dual-V_{th} assignment has emerged as an efficient technique for decreasing leakage power. In Chapter 3, a mixed integer linear programming (MILP) technique simultaneously minimizes the leakage and glitch power consumption of a static CMOS circuit for any specified input to output critical path delay. Using dual-threshold devices, the number of high-threshold devices is maximized and a minimum number of delay elements are inserted to reduce the differential path delays below the inertial delays of the incident gates. The key features of the method are that the constraint set size for the MILP model is linear in the circuit size and a power-performance tradeoff is allowed. Experimental results show 96%, 28% and 64% reductions of leakage power, dynamic power and total power, respectively, for the benchmark circuit C7552 implemented in 70nm BPTM CMOS technology.

Due to the exponential relation between subthreshold current and process parameters, such as the effective gate length, oxide thickness and doping concentration, process

variations can severely affect both power and timing yields of the designs obtained by the MILP formulation. In Chapter 4, we propose a statistical mixed integer linear programming method for dual-V_{th} design that minimizes the leakage power and circuit delay in a statistical sense such that the impact of process variation on the respective yields is minimized. Experimental results show that 30% more leakage power reduction can be achieved by using the statistical approach when compared with the deterministic approach that has to consider the worst case in the presence of process variations.

Compared to subthreshold leakage, dynamic power is less sensitive to the process variation due to its linear dependency on the process parameters. However, the deterministic technique discussed in Chapter 3, which uses path balancing to eliminate glitches, becomes ineffective when process variation is considered. This is because the perfect hazard filtering conditions can easily be destroyed even by a small variation in some process parameters. We present a statistical MILP formulation to achieve a process-variation-resistant glitch-free circuit in Chapter 5. Experimental results on an example circuit prove the effectiveness of this method.

7.2 Future Work

Some ideas and suggestions for future work are given in this section.

7.2.1 Gate Leakage

In this work, the contribution of the gate-tunneling effect to the total leakage is not considered. Neglecting such effect can result in an underestimation of the total leakage. Our examples use BPTM 70nm technology, which is characterized by BSIM 3.5.2 and may not correctly model gate leakage. However, with appropriate design, the gate

leakage of transmission–gate delay elements can be kept small. For example, it is possible to use high-threshold transistors in the delay elements because these transistors are always on and the switching speed is not important. These transistors have a thicker gate oxide layer and hence have a lower gate leakage than low-threshold transistors. Otherwise, in general, the problem of gate leakage will have to be answered by future research.

7.2.2 Techniques for Glitch Elimination with Process Variation

Although leakage has become a dominant contributor to the total power consumption with the continued technology scaling, its contribution drops much lower than the dynamic power after the circuit is optimized by efficient techniques, such as dual-V_{th} assignment and adaptive body bias [11, 13, 28, 54, 63, 90]. Elimination of glitches in a high activity circuit is still imperative. Path balancing is not preferred due to its sensitivity to the process variation. Hazard filtering (gate sizing) is sort of resistant to the process variation but has its own limitation in that a 100% glitch reduction is not guaranteed because of the impossibility of increasing any gate delays on critical paths [8]. Besides, there exists an upper bound on the achievable gate delay in any specific technology. Combining the two methods together to achieve both a complete glitch reduction and a process-variation-resistant circuit should be a challenging topic. In Chapter 5, we propose such a combined technique but at a cost of leakage increase. More efficient algorithms should be developed.

7.2.3 Improvement of the MILP formulation

We have applied our MILP formulation of dual-V_{th} assignment to some industry circuits. (This work was done in the CAD group, Analog Devices Inc., during the summer of 2006).

The basic steps were as follows.

1. Assign all cells in the circuit with low V_{th}. Then the LVT (low V_{th}) delay and leakage for each cell is extracted by PrimeTime [4] from this LVT design.
2. Similarly, acquire the HVT (high V_{th}) delay and leakage for each cell from a HVT design.
3. Extract timing (slack, specified clock period, input-delay, output-delay), primary inputs and primary outputs for each timing group by PrimeTime [4].
4. Construct an MILP model based on the above information.
5. Solve this MILP problem and give the optimal dual-V_{th} solution.
6. Update the circuit from the original LVT design to the dual-V_{th} design according to the CPLEX solution.
7. Check timing and power of the new dual-V_{th} design by PrimeTime [4] and PrimePower [5] respectively.

Experimental results show that twice the leakage power reduction can be achieved by our dual-V_{th} assignment MILP model as compared to a design by commercial tools, Physical Compiler [3] and Astro [2]. About 42% of 15,500 combinational cells were assigned high V_{th}. The runtime for solving this MILP was only several minutes since in such an ASIC design, only small combinational logic clouds (sub-circuits) are inserted

between registers, primary inputs and registers, and registers and primary outputs. Thus, the runtime of an MILP actually depends on the circuit structure of the most complicated or deepest combinational cloud, instead of on the total number of the cells in the circuit.

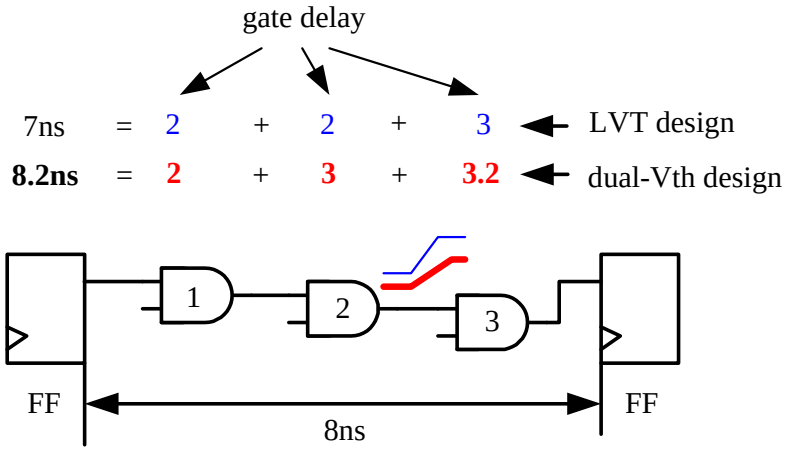


Figure 7.1 An example circuit used for illustrating the timing violation.

However, there are some timing violations (the actual path delay is larger than the timing specification) in the dual-V_{th} design optimized by our MILP formulation. A possible reason is that the delays of LVT cells extracted in step 1 are not accurate. We use the example circuit in Figure 7.1 to briefly explain the cause of a timing violation. In LVT design, the path delay is 7ns (2+2+3) which is less than the specified clock period of 8ns. CPLEX finds that to reduce leakage, gate 2 can be assigned high V_{th} without a timing violation since gate 2's HVT delay is 3ns and hence the new path delay should be 8ns (2+3+3). However, we found that in the dual-V_{th} design, the LVT delay of gate 3 actually changes to 3.2ns due to the increase in its input transition time, as a result of the increase in gate 2's output transition time. Therefore, the real path delay is 8.2ns (2+3+3.2) which is beyond the specified clock period 8ns. The cause of this phenomenon

is the interdependency of delays of gates, which was neglected for simplicity in our MILP formulation.

An iterative method shown in Figure 7.2 may be adopted to get the accurate delays and hence avoid the timing violation problem. If any timing violation is found, the new delays for all LVT cells are extracted from the current dual-V_{th} design and the MILP formulation is updated correspondingly. A different optimal solution is then given by the CPLEX solver with fewer timing violations. We continue iterations until all timing violations are eliminated.

7.2.4 Complexity of the MILP formulation

As discussed in Section 6.3, for a several-million-gate SOC, MILP constraints can be generated for its submodules at a lower level and the run times will be determined by the number of gates in the individual submodules. Such a technique may not guarantee a global optimization, but still would obtain a reasonable result within acceptable run time.

To further reduce runtime of an MILP or ILP formulation, we may also adopt a relaxed LP that uses the LP solution as the starting point and round off the variables such that they satisfy the (M)ILP. Kompella et al. in [48, 49] use branch-and-bound methods to do exhaustive search in the integer space. Although given enough computing time, those methods can find an optimal solution, feasible non-optimal solutions with acceptable run time can be achieved. In [41], the authors propose a new recursive rounding approach which can produce solutions that are close to optimal and, most importantly, the complexity of the new approach is polynomial.

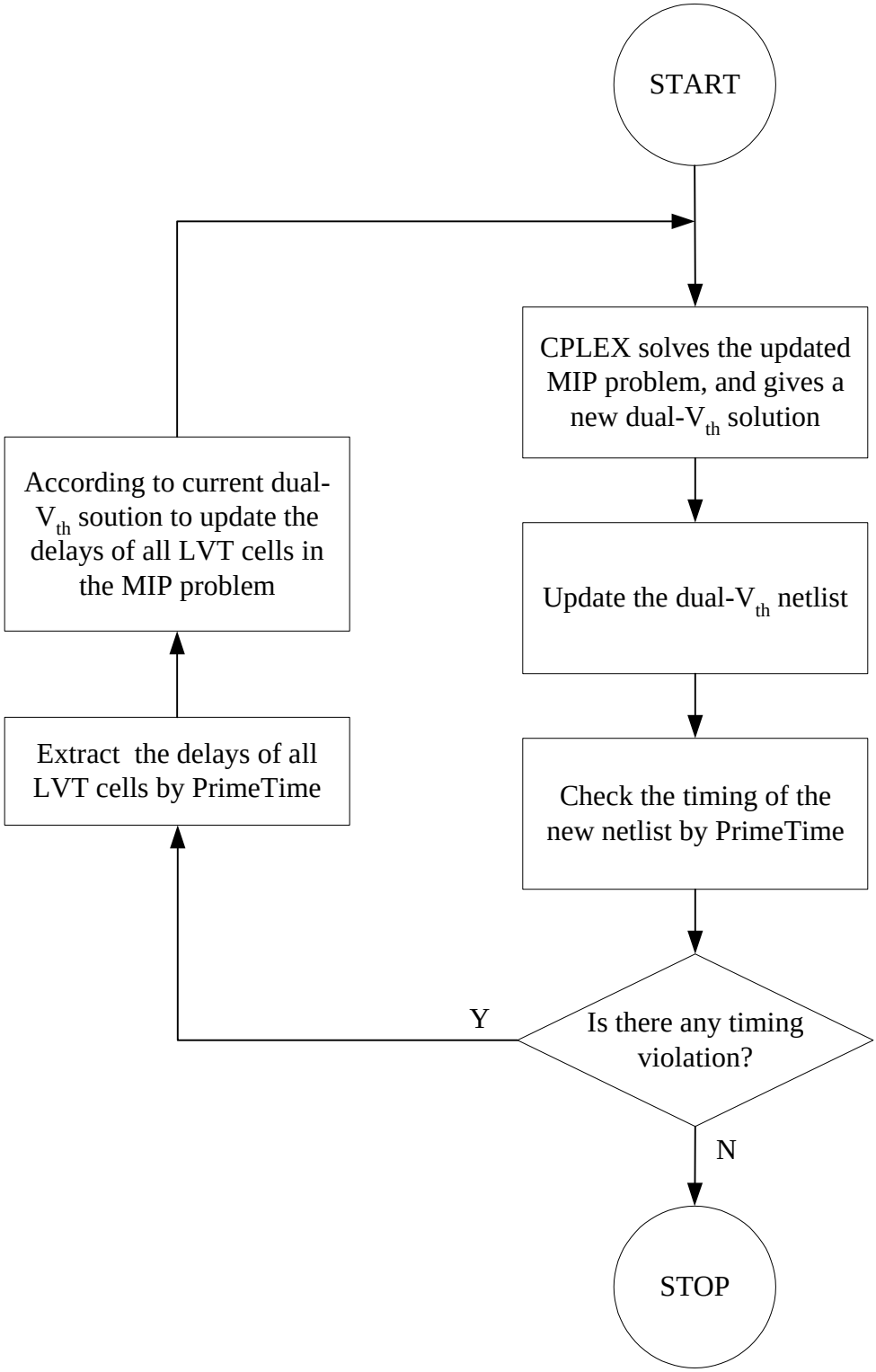


Figure 7.2 Flowchart of an iterative power optimization procedure.

BIBLIOGRAPHY

[1] *BPTM: Berkeley Predictive Technology Model. http://www-device.eecs.berkeley.edu/~ptm/.*

[2] *http://www.synopsys.com/products/astro/astro.html*.

[3] *http://www.synopsys.com/products/unified_synthesis/unified_synthesis.html*.

[4] *http://www.synopsys.com/products/analysis/primetime_ds.html*.

[5] *http://www.synopsys.com/products/solutions/galaxy/power/power.html*.

[6] A. Abdollahi, F. Fallah, and M. Pedram, "Leakage Current Reduction in CMOS VLSI Circuits by Input Vector Control," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 2, pp. 140-154, 2004.

[7] V. D. Agrawal, "Low Power Design by Hazard Filtering," in *Proc. 10th International Conference on VLSI Design*, 1997, pp. 193-197.

[8] V. D. Agrawal, M. L. Bushnell, G. Parthasarathy, and R. Ramadoss, "Digital Circuit Design for Minimum Transient Energy and a Linear Programming Method," in *Proc. of the 12th International Conference on VLSI Design*, 1999, pp. 434-439.

[9] B. Amelifard, A. Afzali-Kusha, and A. Khadernzadeh, "Enhancing the Efficiency of Cluster Voltage Scaling Technique for Low-Power Application," in *IEEE International Symposium on Circuits and Systems*, 2005, pp. 1666-1669

[10] H. Ananthan, C. H. Kim, and K. Roy, "Larger-than-Vdd Forward Body Bias in Sub-0.5V Nanoscale CMOS," in *Proc. of the International Symposium onLow Power Electronics and Design*, 2004, pp. 8-13.

[11] H. Ananthan, C. H. Kim, and K. Roy, "Larger-than-Vdd Forward Body Bias in Sub-0.5V Nanoscale CMOS," in *Proc. of the 2004 International Symposium on Low Power Electronics and Design*, 2004, pp. 8-13.

[12] M. H. Anis, M. K. Mahmoud, and M. I. Elmasry, "Efficient Gate Clustering for MTCMOS Circuits," in *Proc. of the 14th IEEE International Conference on ASIC/SOC*, 2001, pp. 34-38.

[13] V. K. Arnim, E. Borinski, P. Seegebrecht, H. Fiedler, R. Brederlow, R. Thewes, J. Berthold, and C. Pacha, "Efficiency of Body Biasing in 90-nm CMOS for Low-Power Digital Circuits," *IEEE Journal of Solid-State Circuits,* vol. 40, no. 7, pp. 1549-1556, 2005.

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

[15] Z. Bo, D. Blaauw, D. Sylvester, and K. Flautner, "The Limit of Dynamic Voltage Scaling and Insomniac Dynamic Voltage Scaling," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 13, no. 11, pp. 1239-1252, 2005.

[16] M. Borah, R. M. Owens, and M. J. Irwin, "Transistor Sizing for Low Power CMOS Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 15, no. 6, pp. 665-671, 1996.

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

[18] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Testing*. Boston: Springer, 2000.

[19] B. H. Calhoun, F. A. Honore, and A. Chandrakasan, "Design Methodology for Fine-Grained Leakage Control in MTCMOS," in *Proc. of the International Symposium on Low Power Electronics and Design*, 2003, pp. 104-109.

[20] A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design*. Boston: Kluwer Academic Publishers, 1995.

[21] H. Chang and S. S. Sapatnekar, "Full-Chip Analysis of Leakage Power under Process Variations, Including Spatial Correlations," in *Proc. Design Automation Conference*, 2005, pp. 523-528.

[22] X. Chang, D. Fan, Y. Han, Z. Zhang, and X. Li, "Fast Algorithm for Leakage Power Reduction by Input Vector Control," in *Proc. of the 6th International Conference on ASIC*, 2005, pp. 14-18.

[23] C. Chen and M. Sarrafzadeh, "Simultaneous Voltage Scaling and Gate Sizing for Low-Power Design," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 49, no. 6, pp. 400-408, 2002.

[24] O. Coudert, "Gate Sizing for Constrained Delay/Power/Area Optimization," *IEEE Transactions on VLSI Systems,* vol. 5, no. 4, pp. 465-472, 1997.

[25] O. Coudert, R. Haddad, and S. Manne, "New Algorithms for Gate Sizing: A Comparative Study," in *Proc. Design Automation Conference*, 1996, pp. 734-739.

[26] A. Davoodi and A. Srivastava., "Probabilistic Dual-Vth Optimization Under Variability," in *Proc. of the International Symposium on Low Power Electronics and Design*, 2005, pp. 143-147.

[27] M. Elgebaly and M. Sachdev, "Efficient Adaptive Voltage Scaling System through On-Chip Critical Path Emulation," in *Proc. of the 2004 International Symposium on Low Power Electronics and Design*, 2004, pp. 375-380.

[28] W. Elgharbawy, P. Golconda, A. Kumar, and M. Bayoumi, "A New Gate-Level Body Biasing Technique for PMOS Transistors in Subthreshold CMOS Circuits," in *IEEE International Symposium on Circuits and Systems*, 2005, pp. 4697-4700.

[29] A. Forestier and M. R. Stan, "Limits to Voltage Scaling from the Low Power Perspective," in *Proc. of the 13th Symposium on Integrated Circuits and Systems Design*, 2000, pp. 365-370.

[30] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*. South San Francisco, California: The Scientific Press, 1993.

[31] F. Gao and J. P. Hayes, "Total Power Reduction in CMOS Circuits via Gate Sizing and Multiple Threshold Voltages," in *Proc. Design Automation Conference*, 2005, pp. 31-36.

[32] F. Gao and J. P. Hayes, "Exact and Heuristic Approaches to Input Vector Control for Leakage Power Reduction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 11, pp. 2564-2571, 2006.

[33] M. Hashimoto, H. Onodera, and K. Tamaru, "A Power Optimization Method Considering Glitch Reduction by Gate Sizing," in *Proc. of the International Symposium on Low Power Electronics and Design*, 1998, pp. 221-226.

[34] F. Hu, "Process-Variation-Resistant Dynamic Power Optimization for VLSI Circuits," PhD Thesis, Auburn, Alabama: Auburn University, May 2006.

[35] F. Hu and V. D. Agrawal, "Dual-Transition Glitch Filtering in Probabilistic Waveform Power Estimation," in *Proc. of the 15th Great Lakes Symposium on VLSI*, 2005, pp. 357–360.

[36] F. Hu and V. D. Agrawal, "Enhanced Dual-Transition Probabilistic Power Estimation with Selective Supergate Analysis," in *Proc. of the 23rd International Conference on Computer Design*, 2005, pp. 366–369.

[37] F. Hu and V. D. Agrawal, "Input-Specific Dynamic Power Optimization for VLSI Circuits," in *Proc. of the International Symposium on Low Power Electronics and Design*, 2006, pp. 232-237.

[38] E. Jacobs and M. Berkelaar, "Using Gate Sizing to Reduce Glitch Power," in *Proc. of the PRORISC/IEEE Workshop on Circuits, Systems and Signal Processing*, 1996, pp. 183-188.

[39] E. Jacobs and M. Berkelaar, "Gate Sizing Using A Statistical Delay Model," in *Proc. Design, Automation and Test in Europe Conference and Exhibition*, 2000, pp. 283-290.

[40] M. C. Johnson, D. Somasekhar, C. Lih-Yih, and K. Roy, "Leakage Control With Efficient Use of Transistor Stacks in Single Threshold CMOS," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 1, pp. 1-5, 2002.

[41] K. R. Kantipudi and V. D. Agrawal, "A Reduced Complexity Algorithm for Minimizing N-Detect Tests," in *Proc. of the 20th International Conference on VLSI* Design, 2007, pp. 492-497.

[42] J. T. Kao and A. P. Chandrakasan, "Dual-Threshold Voltage Techniques for Low-Power Digital Circuits," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 7, pp. 1009-1018, July 2000.

[43] W. H. Kao, N. Fathi, and L. Chia-Hao, "Algorithms for Automatic Transistor Sizing in CMOS Digital Circuits," in *Proc. Design Automation Conference*, 1985, pp. 781-784.

[44] A. Keshavarzi, S. Ma, S. Narendra, B. Bloechel, K. Mistry, T. Ghani, S. Borkar, and V. De, "Effectiveness of Reverse Body Bias for Leakage Control in Scaled Dual Vt CMOS ICs," in *Proc. of the International Symposium on Low Power Electronics and Design*, 2001, pp. 207-212.

[45] M. Ketkar and S. S. Sapatnekar, "Standby Power Optimization via Transistor Sizing and Dual Threshold Voltage Assignment," in *Proc. International Conference on Computer-Aided Design*, 2002, pp. 375-378.

[46] S. Kim, J. Kim, and S. Y. Hwang, "New Path Balancing Algorithm for Glitch Power Reduction," *IEE Proc. of Circuits, Devices and Systems*, vol. 148, no. 3, pp. 151-156, 2001.

[47] P. Ko, J. Huang, Z. Liu, and C. Hu, "BSIM3 for Analog and Digital Circuit Simulation," in *Proc. IEEE Symposium on VLSI Technology CAD*, 1993, pp. 400-429.

[48] S. Kompella, S. Mao, Y. T. Hou, and H. D. Sherali, "Path Selection and Rate Allocation for Video Streaming in Multihop Wireless Networks," in *Proc. Military Communications Conference*, 2006, pp. 1-7.

[49] S. Kompella, S. Mao, Y. T. Hou, and H. D. Sherali, "Cross-Layer Optimized Multipath Routing for Video Communications in Wireless Networks," *IEEE Journal on Selected Areas in Communications, Special Issue on Cross-Layer Optimized Wireless Multimedia Communications*, May 2007.

[50] H. W. Kuhn and A. W. Tucker, "Nonlinear Programming," in *Proc. 2nd Berkeley Symposium on Mathematical, Statistics and Probabilistics*, Berkeley, 1951, pp. 481-492.

[51] V. Liberali, E. Malavasi, and D. Pandini, "Automatic Generation of Transistor Stacks for CMOS Analog Layout," in *Proc. IEEE International Symposium on Circuits and Systems*, 1993, pp. 2098-2101.

[52] Y. Lin and Q. Gang, "A Combined Gate Replacement and Input Vector Control Approach for Leakage Current Reduction," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 2, pp. 173-182, 2006.

[53] M. Liu, W.-S. Wang, and M. Orshansky, "Leakage Power Reduction by Dual-Vth Designs Under Probabilistic Analysis of Vth Variation," in *Proc. of the International Symposium on Low Power Electronics and Design*, 2004, pp. 2-7.

[54] X. Liu and S. Mourad, "Performance of Submicron CMOS Devices and Gates with Substrate Biasing," in *Proc. of the 2000 IEEE International Symposium on Circuits and Systems*, 2000, pp. 9-12.

[55] Y. Liu and Z. Gao, "Timing Analysis of Transistor Stack for Leakage Power Saving," in *Proc. of the 9th International Conference on Electronics, Circuits and Systems*, 2002, pp. 41-44.

[56] Y. Lu and V. D. Agrawal, "Leakage and Dynamic Glitch Power Minimization Using Integer Linear Programming for Vth Assignment and Path Balancing," in *Proc. of the International Workshop on Power and Timing Modeling, Optimization and Simulation*, 2005, pp. 217–226.

[57] Y. Lu and V. D. Agrawal, "CMOS Leakage and Glitch Power Minimization for Power-Performance Tradeoff," *Journal of Low Power Electronics*, vol. 2, no. 3, pp. 378-387, Dec. 2006.

[58] Y. Lu and V. D. Agrawal, "Statistical Leakage and Timing Optimization for Submicron Process Variation," in *Proc. of the 20th International Conference on VLSI Design*, 2007, pp. 439-444.

[59] V. Mahalingam and N. Ranganathan, "A Nonlinear Programming Based Power Optimization Methodology for Gate Sizing and Voltage Selection," in *Proc. IEEE Computer Society Annual Symposium on VLSI*, 2005, pp. 180-185.

[60] N. R. Mahapatra, S. V. Garimella, and A. Tarbeen, "An Empirical and Analytical Comparison of Delay Elements and a New Delay Element Design," in *Proc. IEEE Computer Society Workshop on VLSI*, 2000, pp. 81 – 86.

[61] M. Mani, A. Devgan, and M. Orshansky, "An Efficient Algorithm for Statistical Minimization of Total Power Under Timing Yield Constraints," in *Proc. Design Automation Conference*, 2005, pp. 309-314.

[62] M. Mani and M. Orshansky, "A New Statistical Optimization Algorithm for Gate Sizing," in *Proc. International Conference on Computer Design*, 2004, pp. 272-277.

[63] M. Miyazaki, G. Ono, and T. Kawahara, "Optimum Threshold-Voltage Tuning for Low-Power, High-Performance Microprocessor," in *Proc. IEEE International Symposium on Circuits and Systems*, 2005, pp. 17-20.

[64] S. Mukhopadhyay, C. Neau, R. T. Cakici, A. Agarwal, C. H. Kim, and K. Roy, "Gate Leakage Reduction for Scaled Devices Using Transistor Stacking," *IEEE Transactions on VLSI Systems*, vol. 11, no. 4, pp. 716-730, 2003.

[65] S. Mukhopadhyay and K. Roy, "Accurate Modeling of Transistor Stacks to Effectively Reduce Total Standby Leakage in Nano-Scale CMOS Circuits," in *Proc. Symposium on VLSI Circuits*, 2003, pp. 53-56.

[66] S. Mukhopadhyay and K. Roy, "Modeling and Estimation of Total Leakage Current in Nano-Scaled CMOS Devices Considering the Effect of Parameter Variation," in *Proc. of the International Symposium on Low Power Electronics and Design*, 2003, pp. 172-175.

[67] A. K. Murugavel and N. Ranganathan, "Gate Sizing and Buffer Insertion Using Economic Models for Power Optimization," in *Proc. of the 17th International Conference on VLSI Design*, 2004, pp. 195-200.

[68] 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-State Circuits*, vol. 30, no. 8, pp. 847-854, 1995.

[69] R. Naidu and E. T. A. F. Jacobs, "Minimizing Standby Leakage Power in Static CMOS Circuits," in *Proc. Design, Automation and Test in Europe*, 2001, pp. 370-376.

[70] D. Nguyen, A. Davare, M. Orshansky, D. Chinney, B. Thompson, and K. Keutzer, "Minimization of Dynamic and Static Power Through Joint Assignment of

Threshold Voltages and Sizing Optimization," in *Proc. of the International Symposium on Low Power Electronics and Design*, 2003, pp. 158-163.

[71] S. M. Nowick and D. L. Dill, "Exact Two-Level Minimization of Hazard-Free Logic with Multiple-Input Changes," in *IEEE/ACM International Conference on Computer-Aided Design*, 1992, pp. 626-630.

[72] P. Pant, R. K. Roy, and A. Chatterjee, "Dual-Threshold Voltage Assignment with Transistor Sizing for Low Power CMOS circuits," *IEEE Transactions on VLSI Systems*, vol. 9, no. 2, pp. 390-394, April 2001.

[73] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*. Upper Saddle River, NJ: Prentice Hall, 2003.

[74] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Minimum Dynamic Power CMOS Circuit Design by a Reduced Constraint Set Linear Program," in *Proc. of the 16th International Conference on VLSI Design*, 2003, pp. 527-532.

[75] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Design of Variable Input Delay Gates for Low Dynamic Power Circuits," in *Proc. of the International Workshop on Power and Timing Modeling, Optimization and Simulation*, 2005, pp. 436–445.

[76] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Variable Input Delay CMOS Logic for Low Power Design,," in *Proc. of the 18th International Conference on VLSI Design*, 2005, pp. 596-604.

[77] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Transistor Sizing of Logic Gates to Maximize Input Delay Variability," *Journal of Low Power Electronics*, vol. 2, no. 1, pp. 121–128, Apr. 2006.

[78] N. Ranganathan and A. K. Murugavel, "A Microeconomic Model for Simultaneous Gate Sizing and Voltage Scaling for Power Optimization," in *Proc. International Conference on Computer Design*, 2003, pp. 276-281.

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

[80] R. Rao, A. Srivastava, D. Blaauw, and D. Sylvester, "Statistical Estimation of Leakage Current Considering Inter- and Intra-Die Process Variation," in *Proc. of the International Symposium on Low Power Electronics and Design*, 2003, pp. 84-89.

[81] R. Rao, A. Srivastava, D. Blaauw, and D. Sylvester, "Statistical Analysis of Subthreshold Leakage Current for VLSI Circuits," *IEEE Transactions on VLSI Systems*, vol. 12, no. 2, pp. 131-139, Feb. 2004.

[82] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Feb. 1990.

[83] C. V. Schimpfle, A. Wroblewski, and J. A. Nossek, "Transistor Sizing for Switching Activity Reduction in Digital Circuits," in *Proc. European Conference on Theory and Design*, 1999.

[84] W.-T. Shiue, "Leakage Power Estimation and Minimization in VLSI Circuits," in *Proc. of the International Symposium on Circuits and Systems*, 2001, pp. 178-181.

[85] J. M. Shyu, A. Sangiovanni-Vincentelli, J. P. Fishburn, and A. E. Dunlop, "Optimization-Based Transistor Sizing," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 400-409, 1988.

[86] J. Singh, V. Nookala, L. Zhi-Quan, and S. Sapatnekar, "Robust Gate Sizing by Geometric Programming," in *Proc. Design Automation Conference*, 2005, pp. 315-320.

[87] A. Srivastava, R. Bai, D. Blaauw, and D. Sylvester, "Modeling and Analysis of Leakage Power Considering Within-Die Process Variations," in *Proc. International Symposium on Low Power Electronics and Design*, 2002, pp. 64-67.

[88] A. Srivastava, S. Shah, D. Sylvester, D. Blaauw, and S. Director, "Accurate and Efficient Gate-Level Parametric Yield Estimation Considering Correlated Variations in Leakage Power and Performance," in *Proc. Design Automation Conference*, 2005, pp. 535-540.

[89] A. Srivastava, D. Sylvester, and D. Blaauw, "Statistical Optimization of Leakage Power Considering Process Variations Using Dual-Vth and Sizing," in *Proc. Design Automation Conference*, 2004, pp. 773-778.

[90] M. Sumita, S. Sakiyama, M. Kinoshita, Y. Araki, Y. Ikeda, and K. Fukuoka, "Mixed Body Bias Techniques with Fixed V_t and I_{ds} Generation Circuits," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 60-66, 2005.

[91] J. W. Tschanz, S. G. Narendra, Y. Ye, B. A. Bloechel, S. Borkar, and V. De, "Dynamic Sleep Transistor and Body Bias for Active Leakage Power Control of Microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 1838-1845, 2003.

[92] S. Uppalapati, "Low Power Design of Standard Cell Digital VLSI Circuits," Master's Thesis, New Brunswick, New Jersey: Rutgers University, Oct. 2004.

[93] S. Uppalapati, M. L. Bushnell, and V. D. Agrawal, "Glitch-Free Design of Low Power ASICs Using Customized Resistive Feedthrough Cells," in *Proc. of the 9th VLSI Design and Test Symposium*, 2005, pp. 41-48.

[94] K. Usami and M. Horowitz, "Clutser Vlotage Scaling Technique for Low-Power Design," in *Proc. International Symposium on Low Power Electronics and Design*, 1995, pp. 3-8.

[95] K. Usami, M. Igarashi, F. Minami, T. Ishikawa, M. Kanzawa, M. Ichida, and K. Nogami, "Automated Low-Power Technique Exploiting Multiple Supply Voltages Applied to A Media Processor," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, pp. 463-472, 1998.

[96] Q. Wang and S. B. K. Vrudhula, "Static Power Optimization of Deep Submicron CMOS Circuits for Dual VT Technology," in *Proc. International Conference on Computer-Aided Design*, 1998, pp. 490-496.

[97] L. Wei, Z. Chen, M. Johnson, and K. Roy, "Design and Optimization of Low Voltage High Performance Dual Threshold CMOS Circuits," in *Proc. Design Automation Conference*, 1998, pp. 489-494.

[98] L. Wei, Z. Chen, K. Roy, M. C. Johnson, Y. Ye, and V. K. De, "Design and Optimization of Dual-Threshold Circuits for Low-Voltage Low-Power Applications," *IEEE Transactions on VLSI Systems*, vol. 7, no. 1, pp. 16–24, Mar. 1999.

[99] L. Wei, Z. Chen, K. Roy, Y. Ye, and V. De, "Mixed-Vth (MVT) CMOS Circuit Design Methodology for Low Power Applications," in *Proc. Design Automation Conference*, 1999, pp. 430-435.

[100] L. Wei, K. Roy, and V. K. De, "Low Voltage Low Power CMOS Design Techniques for Deep Submicron ICs," in *Proc. of the 13th International Conference on VLSI Design*, 2000, pp. 24-29.

[101] L. Wei, K. Roy, and C.-K. Koh, "Power Minimization by Simultaneous Dual-Vth Assignment and Gate-Sizing," in *Proc. of the IEEE Custom Integrated Circuits Conference*, 2000, pp. 413-416.

[102] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and System Perspective*, 3rd ed.: Addison Wesley, 2004.

[103] H.-S. Won, K.-S. Kim, and K.-O. Jeong, "An MTCMOS Design Methodology and Its Application to Mobile Computing," in *Proc. of the International Symposium on Low Power Electronics and Design*, 2003, pp. 110-115.

[104] A. Wroblewski, C. V. Schimpfle, and J. A. Nossek, "Automated Transistor Sizing Algorithm for Minimizing Spurious Switching Activities in CMOS Circuits," in *Proc. IEEE International Symposium on Circuits and Systems*, 2000, pp. 291-294.

[105] A. C. H. Wu, N. Vander Zanden, and D. Gajski, "A New Algorithm for Transistor Sizing in CMOS Circuits," in *Proc. European Design Automation Conference*, 1990, pp. 589-593.

[106] Y. Ye, S. Borkar, and V. De, "A New Technique for Standby Leakage Reduction in High-Performance Circuits," in *Proc. Symposium on VLSI Circuits*, 1998, pp. 40-41.

[107] C. Yeh and Y.-S. Kang, "Cell-Based Layout Techniques Supporting Gate-level Voltage Scaling for Low Power," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 6, pp. 983-986, 2001.

[108] B. Yu, "A Novel Dynamic Power Cutoff Technology (DPCT) for Active Leakage Reduction in Deep Submicron VLSI CMOS Circuits," PhD Thesis, New Brunswick, New Jersey: Rutgers, The State University of New Jersey, October 2007.

[109] B. Yu and M. L. Bushnell, "A Novel Dynamic Power Cutoff Technique (DPCT) for Active Leakage Reduction in Deep Submicron CMOS Circuits," in *Proc. of the International Symposium on Low Power Electronics and Design*, 2006.