Simulation Based Power Estimation For Digital CMOS Technologies

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

Jins Davis Alexander

Certificate of Approval:

Victor Nelson
Professor
Electrical and Computer Engineering

Vishwani D. Agrawal, Chair
James J. Danaher Professor
Electrical and Computer Engineering

Adit Singh
James B. Davis Professor
Electrical and Computer Engineering

George T. Flowers
Graduate Dean
Graduate School

Simulation Based Power Estimation For Digital CMOS Technologies

Jins Davis Alexander

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama
Decemeber 19, 2008

Simulation Based Power Estimation For Digital CMOS Technologies

Jins Davis Alexander

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon the request of individuals or institutions and at their expense. The author reserves all publication rights.

Signature of Author

Date of Graduation

Vita

Jins Davis Alexander, son of Mr. T. D. Alexander and Mrs. Mary Alexander, was born in Kottayam, Kerala, India. He did his schooling in Our Own English High School, Dubai, United Arab Emirates. He earned the degree of Bachelor of Technology in Electronics and Communication Engineering from National Institute of Technology, Calicut, India (formerly known as Regional Engineering College, Calicut) in 1999. He joined the graduate programme in Electrical & Computer Engineering at Auburn University in August 2005.

Thesis Abstract

Simulation Based Power Estimation For Digital CMOS Technologies

Jins Davis Alexander

Master of Science, Decemeber 19, 2008
(B.Tech., National Institute Technology of Calicut, 2003)

87 Typed Pages

Directed by Vishwani D. Agrawal

The estimation of power in digital CMOS circuits has become a significant problem, especially for present day semiconductor technologies. Finding a balance between opposing factors of estimation accuracy and computation speed makes the estimation procedures more challenging. An area that has been neglected is the breakdown analysis of various components of power. In this thesis, we discuss algorithms and a tool for a "total" power estimation tool. This tool does an event-driven simulation of vectors, either supplied or randomly generated at user's option. All components of power, namely, dynamic (separate logic and glitch components), leakage, short-circuit and clock power are estimated. Peak, minimum and average values for the given vector set are also determined. For simulation, first delay, node capacitance, and input state specific leakage are determined for each gate in the given technology, temperature and supply voltage, using a circuit-level simulator (Spice) and saved. This gives us the necessary accuracy and speed to the power estimation.

To demonstrate applications of the tool, we examine leakage variation with temperature, variation of short-circuit power with rise time and output load capacitance, and the quadratic reduction in logic transition power. Glitch power is shown to reduce faster

than the quadratic function of voltage because the increased gate inertia suppresses many glitches. Since in any design technique to reduce one component of power, in general, affects other components, such a tool is useful.

We analyze the effect of process variation in estimation of dynamic power dissipation. Taking a novel approach, we model gates with given lower and upper bounds on delays. For given input vectors, we first find logic transitions using zero-delay simulation. Our algorithms then determine the ambiguity (transient) interval during which transitions occur, and the maximum and minimum number of possible transitions. Computation of these for all gates requires a linear-time analysis of each vector-pair. Weighting with node capacitances estimates lower and upper bounds on dynamic power. Results compare favorably with power analysis using Mont Carlo simulation, which requires significantly more computing resources.

Monte Carlo simulation of ISCAS benchmark circuit c880 for 1000 random vectors (999 vector-pairs) demonstrates the advantages of the bounded delay power analysis. Each vector-pair was simulated for 1000 sample circuits. Sample circuits had gate delays varying $\pm 20\%$ about the nominal values for the TSMC025 2.5V CMOS process [1]. For a vector period of 1000 ps minimum power was 1.424 mW and maximum power was 11.598 mW. Monte Carlo simulation runs took 262.75 CPU s on a Sun Sparc Ultra 10 with 4GB shared memory system. Using the same $\pm 20\%$ variability and the same 1000 vectors, the bounded delay analysis obtained a bound (1.35 mW, 11.89 mW) for power in just 0.3 CPU s. Considering that c880 is a small circuit and the impact of process variation on power continues to assume greater importance, this computational efficiency is a strong motivation for using the method developed in this research.

Acknowledgments

I would like to gratefully acknowledge the assistance, encouragement, support, patience and dedication provided to me by my adviser, Dr. Vishwani D. Agrawal, during my stay at Auburn University. I thank Dr. Victor Nelson and Dr. Adit Singh for being on my thesis committee and for their valuable suggestions. I would also like to thank my colleagues Nitin Yogi, Khushboo Sheth, Hillary Grimes, Fan Wang, Kim and Wei Jang for all the helpful discussions throughout this research. Special thanks to all my friends in Auburn for making my graduate study a pleasant experience.

Most importantly, my heartfelt gratitude and thanks to my parents and my sister, whose encouragement and love has always been my strength in achieving my goals.

Support and recognition of my research by the Wireless Engineering Research and Education Center (WEREC) at Auburn University was very helpful in completing this work.

Style manual or journal used Journal of Approximation Theory (together with the style known as "aums"). Bibliograpy follows van Leunen's *A Handbook for Scholars.*

Computer software used The document preparation package TEX (specifically LATEX) together with the departmental style-file `aums.sty`.

Table of Contents

List of Figures xi

List of Tables xiii

1 Introduction 1
1.1 Motivation . . . 2
1.1.1 Separation of power components . . . 2
1.1.2 Process variation . . . 2
1.2 Problem Statement . . . 3
1.3 Original Contributions . . . 4
1.4 Organization of the thesis . . . 4

2 Prior Work on Power Analysis 6
2.1 Introduction . . . 6
2.2 Power Dissipation in CMOS circuits . . . 6
2.2.1 Dynamic power dissipation . . . 6
2.2.2 Short circuit power dissipation . . . 8
2.2.3 Leakage power dissipation . . . 10
2.3 Existing Power Estimation Techniques . . . 11
2.3.1 Simulation based techniques . . . 12
2.3.2 Probabilistic techniques . . . 14
2.3.3 Statistical methods . . . 16
2.3.4 Other work . . . 18
2.4 Process Variation and Power Estimation . . . 18
2.4.1 Delay variation . . . 19
2.4.2 Existing work . . . 20
2.5 Summary . . . 21

3 Power Estimation 22
3.1 Introduction . . . 22
3.2 Event Driven Logic Simulation . . . 22
3.3 Glitch Filtering . . . 24
3.4 Dynamic Power Estimation . . . 25
3.5 Static Power Dissipation . . . 25
3.6 Short Circuit Power Dissipation . . . 27
3.7 Clock Power and Flip Flop Cell Power . . . 30
3.8 Saving Flip Flop Cell Power . . . 31

3.9 Test Power . . . 33
3.10 Summary . . . 36

4 Power Estimation Results 37
4.1 Introduction . . . 37
4.2 Experimental Procedure . . . 37
4.3 Experimental Results . . . 38
4.4 Summary . . . 42

5 Bounded Delay and Dynamic Power Estimation 44
5.1 Introduction . . . 44
5.2 Background and Definitions . . . 44
5.3 Signal Transition Analysis of Bounded Delay Gates . . . 45
5.3.1 Ambiguity intervals . . . 45
5.3.2 Multiple ambiguity regions . . . 48
5.4 Problem Depiction . . . 50
5.5 Maximum Number of Transitions . . . 52
5.6 Minimum Number of Transitions . . . 55
5.7 Dynamic Power Estimation . . . 57
5.8 Summary . . . 58

6 Bounded Delay Analysis Results 59
6.1 Introduction . . . 59
6.2 Experimental Procedure . . . 59
6.3 Benchmark Results . . . 60
6.4 Summary . . . 65

7 Conclusion 67

Bibliography 69

List of Figures

1.1 Monte Carlo analysis of power dissipation in c880 circuit. 999 random vector-pairs were simulated for 1000 circuit samples. 3

2.1 The charge flow for an inverter : (a) dynamic charing of the load capacitance, (b) discharging of load capacitance. 8

2.2 Short circuit current flow during the switching of transistors. 9

2.3 Leakage current components for an inverter. 11

3.1 Short-circuit scenario: $V_i(t)$ is a rising waveform applied to the input of an inverter with $V_o(t)$ the corresponding output waveform. $I_{sc}(t)$ is the short circuit current waveform that peaks at time t_2 when the PMOS transistor goes from linear to saturation region. 28

3.2 (a) A standard D flip flop and (b) D flip flop with clock gating. 32

3.3 (a) A standard scan cell (SFF) and (b) A scan cell with Q output gated with a scan enable signal (SE) ($SE = 1$ is normal mode). 34

3.4 A scan cell with output Q gated with a scan enable Signal (SE) ($SE = 1$ is normal mode) and with clock gating. 34

4.1 Output voltage waveforms for an inverter with a NAND and NOR load in (a) 0.25 micron (b) 90 nm technologies. 40

4.2 Leakage power of c880 in 90nm technology for 1000 random vectors. 41

4.3 Effect of gate sizing on short circuit power dissipation. 42

5.1 Ambiguity regions of a rising signal (a) and a steady state signal (b) respectively. . 45

5.2 A four-input AND gate with delay bounds (2, 4). Shaded regions are ambiguity intervals. 46

5.3 A three-input AND gate depicting multiple ambiguity intervals. 49

5.4 Two-input AND gate transitions. 51

5.5 Effect of modification factor k on the second upper bound. 54

5.6 Filtering of transitions in a two-input AND gate. 55

5.7 Estimating lower bound on output transitions of a 2-input AND gate. 56

6.1 Monte Carlo simulation versus bounded delay analysis for c880. Each point represents one vector-pair. One hundred sample circuits with *nominal* $\pm$ 20% delay variation were simulated and for each vector-pair (a) maximum and (b) minimum power was determined. 61

6.2 Monte Carlo simulation versus bounded delay analysis for c880. Regression graph for average power. 62

6.3 Transition statistics for high-activity gate 1407 in c2670 for a random vector-pair. Bounded delay analysis: (a) delay bounds (7ps, 12ps), $\underline{\text{mintran}} = 0$, $\underline{\text{maxtran}} = 8$, (b) delay bounds (11ps, 33ps), $\underline{\text{mintran}} = 0$, $\underline{\text{maxtran}} = 4$. Histograms were obtained by Monte Carlo simulation. 63

6.4 A comparison of the maximum power distribution for a vector-pair obtained by bounded delay analysis and Monte Carlo simulation for ISCAS '85 benchmark circuits (a) c880 and (b) c5315. The maximum power values are for 1000 random vector pairs. The Monte Carlo simulation used 1000 circuit samples with random delays to find the maximum power for each vector pair. 64

List of Tables

3.1 Power dissipation results for ISCAS'89 benchmark circuit s5378. 33

3.2 Power dissipation in ISCAS'89 benchmark circuit s5378 with scan cells of various types when the circuit is operated in normal mode. 35

3.3 Power dissipation in ISCAS'89 benchmark circuit s5378 with scan cells of various types when the circuit is being tested. 35

4.1 Comparison with SPICE using an INVERTER with a NAND and NOR load. . . . 39

4.2 1-BIT ADDER Simulation 39

4.3 Average power dissipation for simulation of ISCAS Benchmark circuits using 1000 random vectors in 0.25 micron technology at a supply voltage of 2.5 volts. 41

6.1 Per vector energy consumption in picojoule in benchmark circuits for 1000 random vectors by Monte Carlo simulation of 1000 sample circuits and bounded delay analysis. 60

Chapter 1

Introduction

The increasing use of portable and battery operated devices has raised the demand for low power devices. With higher speed and density of CMOS circuits, power dissipation has become a growing concern. Also, newer technologies are changing the balance of power between various dissipation mechanisms. Thus, a more detailed power estimator is required to facilitate low power design and power optimization of CMOS circuits. Information on how the different components of power vary with different technologies is an important factor for any CMOS designer. Thus, one can understand the significance of a tool that can accurately and efficiently separate each power component. However, estimation of nominal power for a particular technology is not sufficient. Manufacturing process variation, that is, variations in threshold voltage, channel length, etc., and also the effect of the environmental variations in power supply and temperature should also be studied. Process variation can cause uncertainty in delay values, which produce device to device differences in power dissipation. Delay variation effect can be strongly seen in dynamic power estimation. The dynamic power consumed by a digital CMOS circuit depends on logic and glitch transitions, the latter being a function of delays in the circuit. We often take fixed (nominal or worst-case) delay values for design and analysis. This is not correct for two reasons. First, the delay of a gate changes depending on signal states, device temperature, power supply fluctuations, and interconnect coupling noise. Second, and this applies to today's nanoscale technologies, there are wider process variations.

1.1 Motivation

1.1.1 Separation of power components

Most existing tools either estimate the total power or are specific to a particular component of power. However, in the process of optimizing circuits for low power, a designer will be interested in knowing the effects of specific design techniques on each component of power. Different power dissipation mechanisms often have opposing requirements such that a reduction in one power component can simultaneously increase another component. Thus, it is important to provide the designer with separate information about the effect on each of these components besides the total power. One of the motivations of this thesis is to discuss the implementation of a single efficient tool that can estimate the various components of power like dynamic, leakage, short circuit and clock power and then estimate the effect of each component on the overall total power.

1.1.2 Process variation

A second objective in this thesis, is to derive new techniques to estimate dynamic power in the presence of process variability caused variation in gate delays. Currently, a common approach is to use Monte Carlo simulation of sample circuits taking into account the variation in delays. However, this is a time consuming approach. Thus, the need for a faster approach is evident. This can be easily observed from the histogram in Figure 1.1. It shows the Monte Carlo simulation of ISCAS benchmark circuit c880 [2] for 1000 random vectors (999 vector-pairs). Each vector-pair was simulated for 1000 sample circuits. Sample circuits had gate delays varying about the nominal values. The details of technology, process variation, and computing platform are discussed in Section 6. For a vector period of 1000

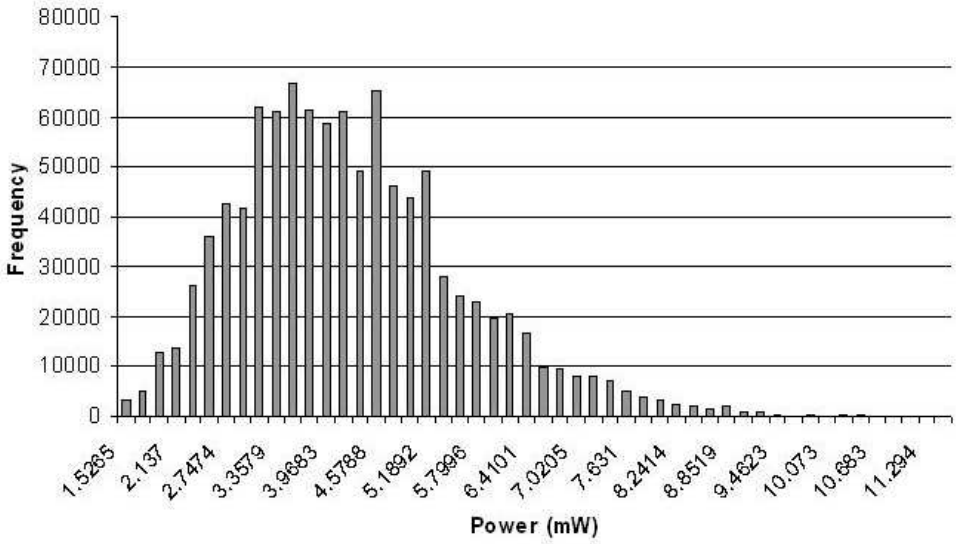

Figure 1.1: Monte Carlo analysis of power dissipation in c880 circuit. 999 random vector-pairs were simulated for 1000 circuit samples.

picoseconds minimum power was 1.424 mW and maximum power was 11.598 mW. Monte Carlo simulation runs took 262.75 CPU s. Using the same variability, the bounded delay analysis developed in the present work obtained a bound (1.35 mW, 11.89 mW) in just 0.3 CPU s. Considering that c880 is a small circuit and the impact of process variation on power continues to assume greater importance, this computational efficiency is a strong motivation for the present work.

1.2 Problem Statement

The problem solved in this thesis: *Find a method to accurately and efficiently estimate and separate the different power components for digital CMOS technologies. Also, develop an algorithm that uses the bounded delay model for dynamic power analysis and thus takes into account manufacturing process variability in gate delays.*

1.3 Original Contributions

We have developed a gate level power estimation tool that can accurately and efficiently estimate various power components as well as the total power and provide useful information to a designer. The tool first computes SPICE based libraries from which it extracts important information for power estimation. Each power component is accurately separated so as to provide useful information for power optimization. For example, the tool can provide maximum and minimum leakage vectors for a particular circuit which can be used to keep the circuit on a standby mode.

To deal with the variability, significant advances have been made in logic simulation, timing analysis and delay testing areas through the use of the *bounded delay* [82] model. The delay of a gate is expressed as a range, typically known as (min, max). In this work, we adopt the bounded delay model to develop a dynamic power analysis method. Although we have only focused on the delay variations, other parameters such as capacitance and leakage current may also be similarly treated in the future. The analysis may also be extended to include leakage power.

1.4 Organization of the thesis

We give a brief introduction to power analysis in Chapter 2. We survey the existing power estimation techniques and the research done in this area as found in the literature. Chapter 3 discusses in detail the power estimation techniques we have used in implementing our gate level power estimator. In Chapter 4, we describe the new bounded delay analysis algorithm to estimate dynamic power in the presence of delay uncertainties. The concepts of bounded delays, ambiguity intervals and multiple ambiguity regions are discussed in detail.

We then give techniques to estimate the maximum and minimum number of transitions on a gate node of the circuit and use them to provide bounds on dynamic power estimation. We show the power analysis results of our gate level power estimator tool in Chapter 5. The results include verification through comparison against HSPICE results on benchmark circuits and some experimental analysis. In Chapter 6. we demonstrate the efficiency of the new bounded delay analysis approach in comparison to the conventional Monte Carlo simulation. Results include analysis of benchmark circuits. We conclude this work in Chapter 7 where we propose future work.

Chapter 2

Prior Work on Power Analysis

2.1 Introduction

We begin with a review of the different mechanisms of power dissipation. We discuss existing power estimation techniques and specify their limitations, which inspired our work. The existing methodologies to incorporate variation while estimating power are also discussed. Our focus is on digital CMOS circuits.

2.2 Power Dissipation in CMOS circuits

There are three main sources of power dissipation in CMOS circuits:

- Dynamic power
- Short circuit power
- Leakage power

2.2.1 Dynamic power dissipation

The dynamic power dissipation is defined as the power spent in charging or discharging of the nodal capacitances during a high to low or low to high transition at the output node. The nodal capacitance consists of the internal capacitance of the gate, the interconnect wire capacitance of the fanout net and the gate capacitances of the corresponding fanout gates.

The power consumed is given by [30]:

$$P_{dyn} = \frac{1}{2}\alpha C_{load} {V_{dd}}^2 f$$

where

- P_{dyn}: dynamic power dissipation of the gate
- α: activity factor of the gate
- C_{load}: load capacitance of the gate
- V_{dd}: supply voltage
- f: clock frequency

The activity factor α determines the amount of switching activity per clock period at the gate output. It lies in the range, $0.0 \leq \alpha \leq 1.0$, with clock ($\alpha = 1.0$) being the most active signal. It determines the amount of switching capacitance, which is an important factor in determining the dynamic power component. In general, the activity factor for a gate depends on the input vectors.

The intrinsic gate capacitances, included in the total load capacitance, are non-linear functions of voltages applied to the devices [21, 45] and are dependent on the region of operation of the transistors. The dynamic energy consumed is due to the charging and discharging of the load capacitance. Figure 2.1 depicts the charging/discharging of the output capacitance C_L of an inverter for a falling/rising input V_{in} and a rising/falling output voltage V_{out}, respectively. The dynamic component can be separated into *logic* power and *glitch or hazard* power. After inputs of a combinational logic block change, the output

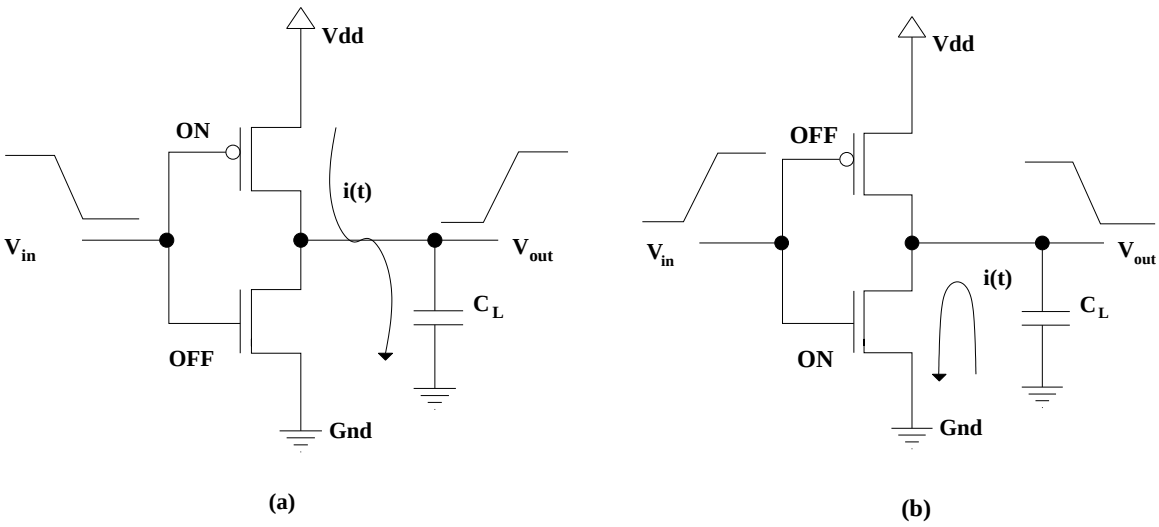

Figure 2.1: The charge flow for an inverter : (a) dynamic charing of the load capacitance, (b) discharging of load capacitance.

of a gate in the block may undergo a series of transitions before reaching its final steady state value. At most a single (i.e., one or zero) transition is necessary to reach the final value. This is known as the *logic transition* and contributes to the logic power component. The remaining unnecessary transitions due to unbalanced path delays in the circuit are called *glitches or hazards* and contribute to the glitch power. It is useful to know these two components of the dynamic power separately for several reasons. For a given circuit, the logic transitions set a lower bound on dynamic power, which can only be affected by changes in the logic design. Glitch power, on the other hand, depends on the physical design, path delays, etc., and can be reduced by other design techniques [4, 5, 74, 75, 76, 77, 78, 91, 92].

2.2.2 Short circuit power dissipation

A short circuit current flows during the period in which the transistors are switching state and are potentially in the "ON" state, thus providing a conducting path from the

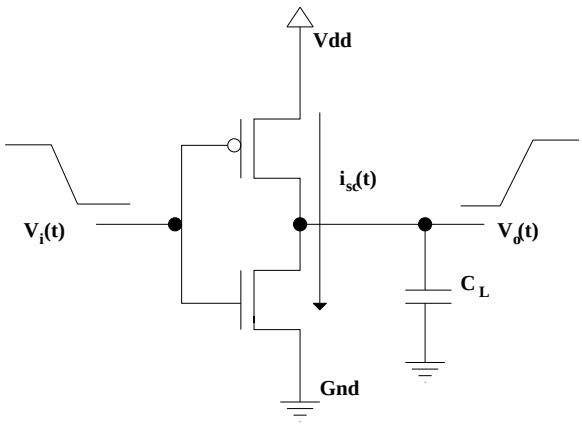

Figure 2.2: Short circuit current flow during the switching of transistors.

supply to ground. This is equivalent to momentarily shorting the supply and ground rails. Fortunately, this happens for a very short duration within the rise or fall time of the input vector signals. The short circuit power dissipation also depends on the output loading capacitances, as they determine how much current will flow from the supply voltage to charge or discharge the capacitance. The remaining current that flows will be the short circuit current, and hence a larger output node capacitance would reduce the overall short circuit current. Thus the short circuit current is at a maximum for low capacitive loads, that is, when the output rise/fall times are much smaller than the input signal rise/fall times. Figure 2.2 shows the short circuit current path by a downward arrow. We will discuss this in greater detail in Section 3.6 where Figure 3.1 depicts the short circuit current waveform $I_{sc}(t)$ as a function of the input $V_i(t)$. The short circuit current peak I_{scmaxf} occurs when the PMOS transistor is going from the linear region to saturation. It is clear from this that for longer input rise/fall times the total average short circuit current flow would be higher. The short circuit current model used for estimation will be discussed in Section 3.6.

2.2.3 Leakage power dissipation

Not too long ago, the leakage power was neglected as being insignificant when compared to the dynamic power consumption. However, for nanoscale CMOS technologies now in use, leakage power has become a significant component of the total power. The leakage power can be attributed mainly to reverse biased pn junctions, the subthreshold leakage current and the gate tunneling effect (Figure 2.3).

The reverse biased pn junction current is the static dissipation due to reverse biased diode leakage between the diffusion regions, wells and substrate [97]. In a submicron process, this component is quite small compared to subthreshold and gate leakages.

Even when a transistor is in the OFF state, a weak inversion current still exists. This is known as the subthreshold leakage. It is exponentially dependent on the threshold voltage and thus, with technology downscaling as the feature size decreases and the supply voltage is scaled down, the threshold voltage also scales down. This results in higher subthreshold current. The subthreshold leakage current is given by:

$$I_{sub} = \mu_0 C_{ox} \frac{W}{L} V_t{}^2 e^{(\frac{V_{gs}-V_{th}}{nV_t})}(1 - e^{\frac{-V_{ds}}{V_t}})$$

where μ_0 is effective mobility, C_{ox} is gate oxide capacitance per unit area, L is the channel length, W is the gate width, V_{gs} and V_{ds} are the gate to source and drain to source voltages, respectively, V_{th} is the threshold voltage, $V_t = \frac{kT}{q}$ is the thermal voltage and n is a technology parameter.

The gate leakage is the oxide tunneling current due to the low oxide thickness and higher electric field. The tunneling current becomes important for technologies where the

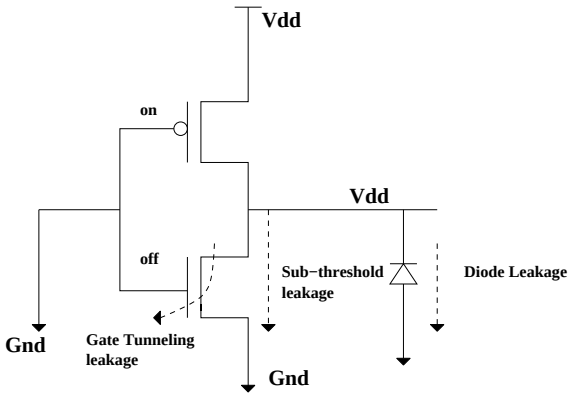

Figure 2.3: Leakage current components for an inverter.

gate oxide thickness is less than $20\mathring{A}$ [97]. This current exists always irrespective of the transistor state.

The leakage power is highly pattern dependent [24] and is dissipated as long as the supply voltage is on. The input vectors applied to the circuit determine the states of the transistors in the steady state condition. Temperature is another important factor. Most devices operate at temperatures higher than the normal room temperature. At these temperatures the leakage power increases substantially, which can be attributed to the increased thermal voltage V_t and the decrease in threshold voltage [50].

2.3 Existing Power Estimation Techniques

Power estimation is an important process of determining the *average or maximum* power consumed by a design, as opposed to instantaneous power which is regarded as a *voltage drop* problem [64]. Without the relevant information one may have to redesign a circuit if it is found to consume more power than expected. In this section, we will discuss the prevalent power estimation techniques. Typically, power estimation techniques

can be broadly classified as dynamic and static [73]. Dynamic methods simulate designs with specific input vector sets and estimate power. Though these techniques are accurate, they are highly time consuming. Static techniques use analytical methods to speed the estimation process at the cost of reduced accuracy.

2.3.1 Simulation based techniques

The most accurate and straightforward approach to power estimation is by circuit simulation using a set of input vectors. The transitions that occur can be easily observed for a gate and can be averaged out for the set of input vectors to give an average power estimate. The advantage of this technique is its accuracy and the fact that it can be used irrespective of circuit, technology or design style. However, it is highly pattern dependent [46, 100], and hence suffers from two major drawbacks. First, it requires extensive use of computing time, especially for large circuits. Second, a designer may not know the set of input patterns when the power for a particular block, embedded in a large system, is to be calculated. Thus, the power calculated may be erroneous as some of the input patterns used for estimation may never occur during normal operation.

Circuit simulators like SPICE [62] use large matrix solutions of Kirchoff's current law (KCL) equations to determine the nodal currents at the transistor level. Basic components like resistors, capacitors, inductors, current sources, voltage sources and higher level device models of diodes and transistors are used to accurately estimate the current, voltage drop and even the non-linear capacitances present at transistor nodes. From this information highly accurate power analysis is possible. The complex device modeling for higher accuracy increases the computation complexity to such a level that SPICE is no longer a feasible

option for larger circuits. To improve on the speed, another transistor level simulator called PowerMill [44] uses piecewise linear transistor modeling to capture the transistor characteristics in lookup tables. It also uses an event driven timing algorithm so as to obtain speeds comparable to logic simulators but with the difference that it does not consider logic transitions but instead changes in node voltages. The use of lookup tables leads to inaccuracies but results in a speed up of 2 to 3 times compared to SPICE.

Switch level simulators like MOSSIM [14, 17, 38] and IRSIM [81] view transistors as bidirectional switches and circuit nodes as charge storage nodes. When a transistor is in an ON state, the switch closes creating a conduction path between the drain and source nodes of a transistor. In this model, simulation can be performed with an approximate RC calculation, thus making it faster than the normal transistor level analysis. Switch level simulators can be extended for power analysis by calculating the approximate switching capacitance for dynamic power estimation [89]. Though other components like leakage and short circuit power can be estimated, these are not very accurate compared to transistor level analysis. For example, short circuit power must be accounted for by examining the time in which the switches form a path from power to ground. A switch level simulator does not accurately model timing. Besides, the modeling does not consider the output load capacitance which leads to further inaccuracies.

Gate level simulation involves the use of logic components like NAND/NOR gates, latches, flip flops and interconnection nets. The most common analysis method involves an event driven simulation [17]. When a transition or *event* occurs at an input of a gate, it may trigger an output event after a certain time delay. Power consumption is estimated by calculating the switching capacitance at the node of the gate, and by the number of

events that occur at that node. However, in this type of analysis, each gate is modeled as a black box and the internal structure is not considered. Thus short circuit current and other internal capacitances are ignored [11]. Cell based power estimation techniques follow similar methods [10] in which libraries of cells are characterized by electrical (SPICE-level) simulation for all possible input combinations and fan-in/fan-out possibilities. Logic simulation uses this information to estimate power. The extent of the accuracy depends on the type of macro models followed and the accuracy of capacitances provided at the cell level [33].

2.3.2 Probabilistic techniques

Probabilistic methods involve modeling transitions occurring at a gate as probability functions. The probabilities of the nodes to change their logic state are propagated through the circuit. Since probabilities are used, no input vectors are required, resulting in a reduction in computational effort. Thus, these techniques are considered as *pattern independent* [15]. By deriving the switching activity as probabilistic measures, coupled with the node capacitances, one can estimate dynamic power. Issues like signal independence, spatial and temporal signal correlations determine the accuracy and complexity of the technique [83]. Also, most of the techniques focus on dynamic power, disregarding other components which can become significant at submicron and nanoscale technologies.

In an early work of probabilistic estimation [27], *zero delay* model was assumed for all gates. *Signal probabilities* (denoted as P_s) were propagated through a circuit using simple probability expressions. For example, a two input AND gate with $y = AND(x1, x2)$ has an output signal probability $P_s(y) = P_s(x1)P_s(x2)$, assuming the input events $x1$ and $x2$

are independent. The *transition probability*, that is, the probability of signal transition (denoted as P_t) is calculated under the assumption of temporal independence as $P_t(y) = 2P_s(y)(1-P_s(y))$. The disadvantages of this method are, first, the use of a zero delay model neglects glitches, and second, both temporal and spatial independences introduce further inaccuracies.

Probability waveforms were used in simulators like CREST and other techniques [15, 66, 67]. A signal probability waveform gives the probabilities for signal values and transitions during a vector period. It is a sequence of signal probability values for intervals between the instants of rising and falling transitions. The probability waveforms are propagated through the circuit. The propagation algorithm is similar to event driven simulation with one difference that instead of propagating exact transitions, the probability of a transition is propagated. Here spatial correlations have not been considered. These have been addressed in works describing tagged probabilistic simulation (TPS) [29, 90]. Hu and Agrawal [41, 42] developed a new glitch filtering method using dual transition probabilities and they observed that by using these techniques along with tagged probabilistic simulation more consistent power estimation was obtained.

A useful concept is *transition density* proposed by Najm [63]. Transition density is defined as the average number of transitions for a particular node in one clock period. The propagation of transition density functions is done by *Boolean difference* [17]. If y is a Boolean function that depends on x, then the Boolean difference is expressed as,

$$\frac{\partial y}{\partial x} \triangleq y|_{x=1} \oplus y|_{x=0}$$

where $\oplus$ denotes exclusive-OR function. If the inputs to a Boolean module, x_i, are assumed to be spatially independent, then the density of its output y is given by,

$$D(y) = \sum_{i=1}^{n} P(\frac{\partial y}{\partial x}) D(x_i)$$

where transition density is denoted as D. However, the propagation of transition density here assumes that no two transitions occur at the same time. Also, the calculations of Boolean difference get more complex for signals nearer to *primary outputs* (POs) of a circuit.

The above techniques so far have used only deterministic delay models, meaning that the transition probability waveforms consider the probabilities of transitions occurring at discrete time intervals based on fixed gate delays. With technology scaling, delay variations and uncertainties have become important considerations. In a recent work [32], authors proposed an algorithm of propagating transition probability waveforms considering uncertainty in delays. The probability waveforms described as continuous functions of time provided more accurate results as compared to fixed delay techniques.

2.3.3 Statistical methods

Statistical techniques have been explored where randomly generated input patterns are applied to the circuit and monitored using a simulator until a desired accuracy of power estimation is achieved. The method is known as *Monte Carlo* analysis because the circuit is simulated repeatedly using a logic simulator, monitoring the power until it converges close to an average. To decide on how to select the input patterns so that the power converges, various statistical mean estimation techniques are employed. McPOWER [16] is the first known Monte Carlo power estimator. It is a variable delay simulator, which

measures power from random sets of input vectors and from the resulting data determines the average. The process is continued until the mean is considered stable for a large set of patterns. One can see that this method is time consuming but gives better accuracy as compared to probabilistic methods.

Xakellis and Najm [99] improved upon McPOWER by considering transition densities to come up with *mean estimator of density* (MED). They statistically estimate the individual node transition densities. It would be inefficient to let statistical estimates converge for all gates, especially as it would take a large number of vectors to converge for those nodes that switch infrequently. As a result, the authors classified nodes as *low density* and *regular density* nodes, based on a threshold value. A major advantage of this technique is that the error tolerance levels can be specified by users upfront and nodes with least tolerance can also be identified. The method was improved upon [70] by using different variable errors for different nodes, unlike the use of a constant user - specified estimation error. This way variable error rates are determined for nodes with higher switching activity in such a way as to estimate them more accurately. Other works of statistical nature include estimating power under uncertainties in input vector specifications. Since power is highly dependent on vector patterns applied, any uncertainties in their specifications can make the estimation process difficult. This problem has been addressed by considering average power as a range (minimum average power, maximum average power) and statistically estimating the sensitiveness of average power dissipation with respect to the uncertain nature of input vectors [26].

2.3.4 Other work

We have so far discussed notable techniques for power estimation. Other works include those which concentrate on specific power components. PowerPlay [49] is a fast algorithm to speed up the estimation of dynamic power. It is essentially a logic simulator that calculates the instantaneous power waveform at a gate (or cell) level. In other works [60, 79], leakage power estimation is emphasized where, based on the characterization of the device state, both pattern dependent and independent techniques are employed. Some works include analytical modeling of leakage current [61], considering variations in process parameters like doping profile, flat band voltage and supply voltage. In [25], the authors discuss accurate transistor stack modeling so as to improve efficiency by reducing the dependence of the device state on all possible input combinations. Significant work has also been done on developing accurate short circuit models [8, 39, 93, 96]. These have been extended to logic gates [94] so as to improve the accuracy of estimation for gates other than standard inverters.

2.4 Process Variation and Power Estimation

Process variations are defined as the variations in the semiconductor fabrication process causing changes in threshold voltage, oxide thickness, channel length, interconnect wire width, thickness, etc. Process variations are divided into inter-die and intra-die variations. Inter-die variations are the variations among dies on the same wafer (or wafer lot), while the parameters remain constant within the same die. Intra-die variations, however, are differences within the same die, that is, devices within the same die have variations in certain parameters. It has been observed that intra-die variations have spatial correlations. This

means that those devices that are in close proximity within a die have a higher probability of being alike than devices that are physically far apart.

In the present research, we have considered only process variations in gate delays, and their effect on dynamic power consumption. In reality, both inter-die and intra-die variations can effect load capacitances. However, since there can be increase on some nodes and decrease on others, this source of variation on an average may not lead to an increase in power dissipation in an optimized circuit. In the case of gate delays, an increase/decrease in delays can cause unbalanced paths or change the inertial delay properties of gates. This on the whole can change the switching activity of a circuit, leading to changes in total power dissipation.

2.4.1 Delay variation

Circuit delay is particularly sensitive to process variations because it is dependent on a number of other variation-sensitive parameters. Variation in delay also adversely changes the dynamic power dissipation, because depending on the variation and topology of the circuit, it may or may not increase the number of transitions. That is why its effect is of utmost importance in power estimation.

A simple model of gate delay can be given as ([21], p. 89),

$$T_d = \frac{C_L \times V_{dd}}{I} = \frac{C_L \times V_{dd}}{\frac{\mu C_{ox}}{2}\frac{W}{L}(V_{dd} - V_t)^2}$$

where C_L is the node capacitance, V_{dd} is the supply voltage, V_t is the threshold voltage, $C_{ox} = \frac{\epsilon_{ox}}{T_{ox}}$ is the gate oxide capacitance per unit area and T_d is the gate delay. From

this equation it can be seen that, even for a small change in the dependent variables, the combined effect of all the variations can cause a significant change in delay.

The effect of temperature and variation in supply voltage can also change the delay. In this work, we only consider process variations and neglect other effects.

2.4.2 Existing work

Effects of process variation on power has been discussed [22, 28, 61, 85, 86, 87] mainly in the analysis, optimization and reduction of leakage power. Another way to deal with process variation is to use a Monte Carlo approach [95]. Here we take the variations between individual dies into account and consider the delay to be a random variable. We then run Monte Carlo simulations to get a high quality simulated maximum power sample from which by statistical techniques we estimate the exact and mean maximum power. Due to the number of simulations required to have a reasonably accurate estimate, this technique is quite time consuming. This was also used in a system level power analysis methodology as discussed in [20]. Here the authors used a number of techniques including Monte Carlo sampling and power state (idle, sleep, active) leakage modeling while considering process variations to estimate system level power.

Process variation has been considered in logic-level simulation, critical path delay and timing analysis through delay modeling of gates. Both statistical and bounded delay models have been used [7, 52, 87]. In the bounded delay model, each gate is assigned the lower and upper bounds for delays, also called the *min-max* delay specification [12, 13, 18, 19, 37, 54, 80, 82]. In recent work [12, 13, 34, 35] the correlations at the inputs of reconvergent gates were considered to improve the accuracy of bounded gate delay fault simulation. By

ignoring the effect of reconvergent fanouts, the results were pessimistic and the authors were able to improve the quality of the gate delay tests by considering the correlations.

2.5 Summary

In this chapter, we have discussed various existing techniques for power estimation. In general, they have been classified as simulation based and static (non-simulation) approaches. It is to be noted that, in most of the above techniques, focus has been on estimating total or average power consumption. There have been other notable works focusing on a particular power components. However, it can be seen, due to the size of the problem, that there are few works that estimate and separate each component with notable accuracy and efficiency. We also discussed the effect of process variation and the effect of delay uncertainties on power estimation. We have also given a brief description of some existing approaches in this area and the current techniques that use the bounded gate delay approaches.

Chapter 3

Power Estimation

3.1 Introduction

In the previous chapter, we discussed existing and current power estimation techniques. As observed there, the major hurdles in power estimation are the conflicting factors of accuracy and efficiency. Simulation based techniques, though accurate, are time and memory consuming, while probabilistic and statistical approaches reduce the accuracy. It is clear that a lot of work has been done in solving this problem. Most of the tools concentrate on estimating average power efficiently, while few concentrate on specific components. In the following sections, we aim to provide a tool that can accurately and efficiently separate each power component. The motivation is to have a single tool that can do simulation and power analysis, with emphasis on getting as much information as possible about each power component. In the subsequent sections, we will discuss the important methods and techniques we used to achieve this.

3.2 Event Driven Logic Simulation

The first step, before any power analysis method, is to simulate the concerned circuit so as to get as much information about the circuit activity and its response to the given input vectors. This can be easily done by simple logic simulation methods to give us accurate switching activity information. However one may argue that for an accurate average power estimation by simulation methods, the power estimation tool would require a representative set of input vectors that will be too large for big circuits. The time and memory consuming

nature of this method has resulted in alternative methods of probabilistic and statistical nature which improve speed at the cost of reduced accuracy. This may be suitable for dynamic power estimation, but since leakage power is also input specific, the overall reduction in accuracy is significant. Thus, even though it may be time consuming, most tools in the industry do simulation of functional vectors and then dump the waveforms or switching activity information for power analysis.

For simulation of input vectors, we have used the most efficient method of event driven simulation. This type of simulation is at the gate level and is considered a standard technique for logic simulation of circuits. In event driven simulation, events that occur at the gate outputs are scheduled to be processed in the order of their temporal occurrence. When an event occurs at time t at the output of gate it may or may not cause events at its fan out gates. If it does cause an event at a fan out gate, say at a time $t + \Delta$, then this event is scheduled to be processed at that time. This way, only those gates which have output events are processed at their respective time occurrences. It can be seen that event driven simulation is dependent on the gate delays, as they determine when the next events are scheduled. There are two types of delays that are processed during standard logic simulation, namely *transport delays* and *inertial delays*. The transport delay model just delays the change in the output by the time specified by the delay modeled. This is useful for cases of buses which may not absorb short pulses. However real gates have an inertia property which prevents any transitions with pulse width less than the gate delay from propagating. In other words a gate has an inertia to these kind of signal changes and this is modeled accurately by the inertial delay model. Here rather than schedule an event to occur during the next event processing, it is scheduled to occur after applying the delay to the current

time. Thus the simulator must maintain the current time value. When no more events exist to be processed at the current time value, time is updated to the time of the next earliest event and all events scheduled for that time will be processed. We have followed the inertial delay model for gates in our logic simulation, with delays being characterized with a wire load model which uses the fan out information of the gate.

To improve the memory usage, we have implemented a circular stack for event driven simulation. This way events being processed for future time intervals will be pushed to the beginning of the stack, overwriting already processed events. In this way, the need for a huge memory stack is avoided, especially for larger circuits. The analysis is stopped when all events on the stack have been processed indicating the circuit is at a steady state.

3.3 Glitch Filtering

The event driven simulation will simulate all events that will occur at the output of a gate. This means that if a gate has glitches or extra unnecessary transitions before it reaches its final logic state, they will be processed as events in order of their time occurrence. Since we consider the inertial delay properties of a gate, we filter out certain glitches that have too short a pulse width as compared to the gate delay [4]. If this is not addressed one may overestimate the switching activity and thus give the wrong information for power analysis. In our tool we have implemented a glitch filtering algorithm that addresses this and gives accurately the actual transitions that will occur at a gate output.

3.4 Dynamic Power Estimation

Our approach is similar to commercial based cell or library based approaches. During logic simulation, if a gate output node has an event or transition, we dynamically SPICE characterize capacitance libraries for that particular cell. For accurate characterization we include the transistor sizing variations for different technologies, and also the region of operation the transistor is in. We have used both the gate oxide capacitances as well as the internal diffusion capacitances while estimating the overall node capacitance. Using this, for every transition we can estimate the energy spent in charging and discharging the capacitance using the dynamic power dissipation equation, as mentioned in Section 2.2.1. We then average the power over the input vectors to get an average dynamic estimate. Our tool can accurately separate *logic* and *glitch* power, which are illustrated in our results. Though we have neglected the energy dissipated due to internal charging of parasitic capacitances, it can be seen from our results that overall contribution of this energy to the total power is not that prominent.

3.5 Static Power Dissipation

Subthreshold leakage currents can be estimated from a number of BSIM models [88, 84]. It should be noted that the leakage dissipation is highly dependent on the input state the transistors are in. However, to estimate the leakage current for all possible input combinations for a gate will be time consuming. For a faster but accurate estimation process, we have implemented the transistor stacking model as followed in [24, 36]. Here when the transistors are in OFF states and in series, we estimate by SPICE characterization, the leakage current flowing. This is unavoidable as the transistors in series have different

voltages across their drain-source terminals, resulting in different currents (see the transistor stacking model [24, 36]). However if the transistors are in parallel, and in OFF states, all we need is to estimate the current for one equivalent transistor, and multiply by the number of parallel transistors. If any parallel transistor is in an ON state, then this will result in a short circuit, and the defining current is flowing through the ON transistor thus bypassing any leakage current flow in the OFF transistors.

The average leakage estimation for a particular gate can be formulated from [51] as:

$$P_{leak} = \frac{\sum_i (I_{DSqi} \times V_{DSqi} \times t_{inp})}{T_{analysis}} \tag{3.1}$$

where P_{leak} is the average leakage estimate for a gate, I_{DSqi} is the quiescent leakage current for the i_{th} transistor in OFF state, V_{DSqi} is the drain source voltage across the transistor, t_{inp} is the time that the gate is in that particular input state and $T_{analysis}$ is the total analysis period time.

It should be noted that, as pointed out in [9, 51], since leakage estimation is vector dependent it may be time consuming for large circuits. In [51], the authors have come with a leakage regression model based on the number of gate cells obtained from previous experimental data of different smaller circuits. However, this is not feasible as one would have to repeat the acquisition of experimental data for different technologies. In our implementation, for each new technology, leakage libraries are dynamically created. Once created for a particular gate, and for a particular input state, it need not be created again and is just read in when required. This is useful as the same cell may be used for different circuits and need not be created again.

3.6 Short Circuit Power Dissipation

The estimation of short circuit dissipation has been a difficult problem. Most tools usually couple this with the dynamic dissipation and report the *switching* energy. Though this is correct, we aim to separate these two components, as the short circuit current flow has its own interesting characteristics that need to be studied. A number of short circuit models have been proposed and studied [8, 39, 93, 96]. An accurate model takes into account the effect of the loading capacitance as well as the input rise/fall times. The rise/fall time period determines the overall current flow while the loading capacitance determines how much of the short circuit current flows during the charging or discharging of the node capacitance. Thus a larger output load means a decrease in short circuit current. This is illustrated in Figure 3.1 which depicts a case of a rising input waveform applied to an inverter. The NMOS transistor will be in the saturation region while the PMOS transistor will be in the triode region during the time period t_1 to t_2. We can see that the peak short circuit current flows at the time the PMOS transistor moves from the triode region to the saturation region. By estimating this peak current, and calculating the area under the short circuit current curve we can get an estimate for the total short circuit flowing during this period.

The estimation of the peak short circuit is calculated using a model described in [96]. Once again consider the time period t_1 to t_2 in Figure 3.1. The current discharging through the NMOS transistor when it is in the saturation region can be modeled approximately by the differential equation:

$$-\frac{C_L}{V_{dd}}\frac{dv_{out}}{dt} = 0.5K_n(v_{in}-n)^2 \tag{3.2}$$

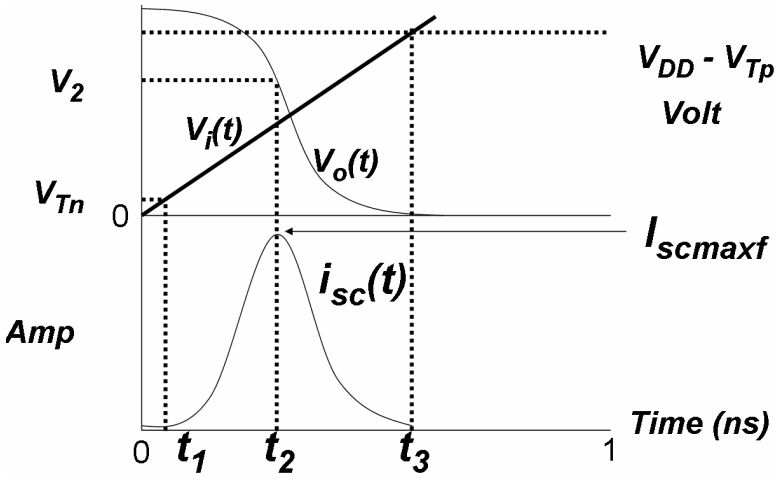

Figure 3.1: Short-circuit scenario: $V_i(t)$ is a rising waveform applied to the input of an inverter with $V_o(t)$ the corresponding output waveform. $I_{sc}(t)$ is the short circuit current waveform that peaks at time t_2 when the PMOS transistor goes from linear to saturation region.

for $v_{in} - n < v_{out}$ and where C_L is the load capacitance, V_{dd} is the supply voltage, v_{out} is the normalized output voltage, K_n is the transconductance factor of the NMOS transistor, v_{in} is the normalized input voltage and n is the normalized threshold voltage. Upon integration the output voltage with respect to time t can be found as [39]:

$$v_{out} = 1 - \frac{V_{dd} K_n t_0}{6 C_L} (\frac{t}{t_0} - n)^3 \tag{3.3}$$

where

$$v_{in}(t) = \frac{t}{t_0} \tag{3.4}$$

where t_0 is the rise time of the input signal and $t \in [0, t_0]$. It can be seen from the above equations that the important component of the current through the PMOS transistor has been neglected. This has been done for ease in solving the above equations. However this

creates inaccuracies in the estimation of short circuit current as the PMOS current is not negligible when compared to the current flowing through the NMOS transistor. This results in underestimating the output voltage and thus an overestimating the short circuit current. To correct this, a correction factor is used by scaling the results from equation 3.3 with actual SPICE simulations. The above equation is then rewritten as:

$$v_{out} = 1 - \frac{1}{6}(\frac{V_{dd}K_n t_0}{C_L})^{\epsilon}(\frac{t}{t_0} - n)^3 \tag{3.5}$$

where ϵ is the correction factor [96]. From our analysis the value of ϵ varies with input rise time t_0, output load capacitance C_L and the effect of the PMOS transistor on the short circuit current which is done by including the transconductance factor K_p. Thus ϵ can be modeled as:

$$\epsilon = \epsilon_0 K_p + \epsilon_1 t_0 + \epsilon_2 C_L + \epsilon_3 \tag{3.6}$$

where ϵ_0,ϵ_1,ϵ_2,ϵ_3 are technology parameters than can be determined from SPICE simulations.

We now estimate the peak current by finding the output voltage at time t_2 (as seen in Figure 3.1) from the above modeled equations. For this purpose we first calculate the time t_2 by replacing $v_{out} = v_{in} - p$ in equation 3.5, as that will be the output voltage just at the time PMOS begins to go to saturation. From the resulting third order polynomial we get one real solution for t_2. Once we estimate the output voltage at time t_2, using the drain current equations we can estimate the current flowing through the PMOS transistor and

then calculate the area under the short circuit current curve to get the total short circuit current flowing during this period.

Thus we can then depict the short circuit energy calculated as:

$$E_{scf} = \int_{t_1}^{t_3} V_{dd} I_{sc}(t)\, dt = (t_3 - t_1) I_{scfmax} \frac{V_{dd}}{2} \tag{3.7}$$

where V_{dd} is the supply voltage, $I_{sc}(t)$ is the short circuit flowing at time t, and I_{scfmax} is the peak short circuit current.

For gates other than inverters, like NAND/NOR, they are converted into equivalent sized inverters and the short circuit current is estimated. Though this an approximately it gives reasonable results. This is understood to be a reasonable approach as parallel transistors can be modeled as equivalent sized transistors without any loss in accuracy. In case of series transistors, complexity increases due to multiple nodes and different regions of operation the transistors are in. However during short circuit estimation, since the series connected transistors are not considered during the charging or discharging phase, this parasitic behavior, as discussed [69], can be modeled with sufficient accuracy with equivalent sized transistor. In cases of more complex gates, they are broken down to simpler gates and and then modeled as equivalent sized inverters [48, 69] to estimate the short circuit current.

3.7 Clock Power and Flip Flop Cell Power

The clock power consumption is considered one of the largest portions of power dissipation. Typically, the clock power accounts for 40% or more [23, 31] of the total processor

dissipation, rivaling only that those dissipated by memory structures. This may be accounted by the fact that the clock is distributed throughout the circuit and results in higher wiring capacitance, it affects the most blocks and thus faces a larger load and also by the fact that two transitions occur every clock cycle. At the gate level, estimating the total load the clock affects including the wiring capacitance, with the clock frequency will provide an apt clock power estimation. Another component that our implementation focuses on is the power dissipated in the flip flop cells [65]. It should be noted that in a flip flop, even when the Q output does not change, power is dissipated due to internal switching within the cell. This switching continues when the D input changes or just the clock keeps switching. We estimate the clock power by calculating the total capacitance that is seen by the clock and then calculate the energy spent in charging and discharging this capacitance per clock cycle. Similarily for flip flop cell power, we characterize the energy spent during the internal switching within the flip flops.

3.8 Saving Flip Flop Cell Power

From our observations, a significant amount of power is consumed in a sequential circuit as clock power and in the flip flop cells as described in the previous section. One way to reduce these components will be to not clock the flip flops if the D input and Q output have the same value. This can be done through a simple *clock gating* approach in which an XOR-AND gate is added to the D flip flop circuitry as shown in Figure 3.2 [71, 101].

We conducted experimental simulation of ISCAS'89 benchmark circuit s5378, which has 179 flip flops. The aim was to compare and observe the effect of clock gating on the various power dissipation components. The results are shown in Table 3.1 where we compare

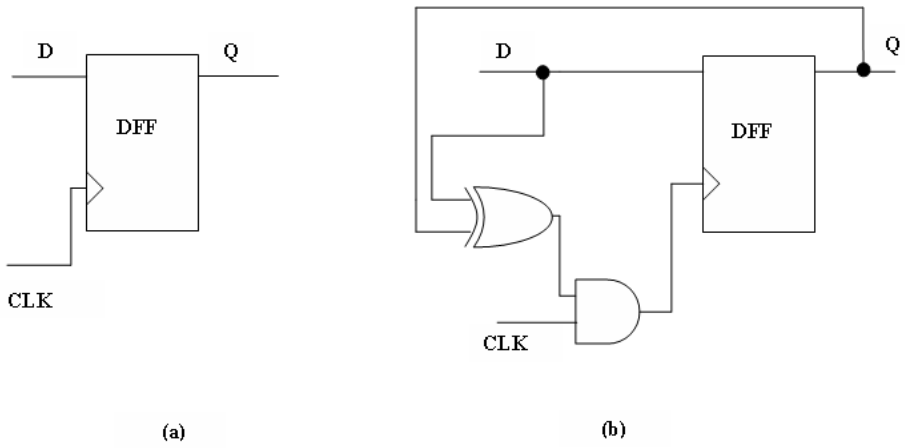

Figure 3.2: (a) A standard D flip flop and (b) D flip flop with clock gating.

the power dissipated by the circuit with normal D - flip flops (Figure 3.2 (a)) and that with D - flip flops with clock gating (Figure 3.2 (b)). In Table 3.1, they are shown as s5378 and $s5378_clk$, respectively. The power results are for 1000 random vectors and a clock period of 50 ns was used. These results are for TSMC025 technology. It is assumed that after the circuit is in steady state, the next vector is applied to the inputs. As can be seen from the results, because there is no unnecessary clocking when the inputs and outputs of the flip flops are identical, there is a significant decrease (from $751.6\mu W$ to $32.5\mu W$) in the flip flop cell power dissipation. We observe that there is a slight increase in dynamic as well as in leakage power components which can be attributed to the overhead of extra gates. This increase in the combinational power is attributed to the fact that for each flip flop input change, there may be a maximum of 6 transitions within the XOR-AND gate combination. On balance, from the data in the last column of the table, we see a 71.5% reduction in power dissipation.

Table 3.1: Power dissipation results for ISCAS'89 benchmark circuit s5378.

Circuit name	No. of gates	Logic power (μW)	Glitch power (μW)	Dynamic power (μW)	Short Ckt. power (μW)	Leakage power (μW)	Clock power (μW)	Flip Flop power (μW)	Total power (μW)
$s5378$	2958	77.9	17.46	95.40	14.09	0.1291	220.26	751.60	1081.47
$s5378_clk$	3316	79.23	54.22	133.46	23.06	0.1329	118.88	32.50	308.02

3.9 Test Power

In this section, we will examine the dissipation of power during the scan testing of a circuit, which is a popular method for testing of sequential circuits [17]. This subject has received much attention in recent years [68]. Test power has two components, *shift* power and *capture* power. The shift power is the power consumed during the shifting of vectors through scan cells. Since the outputs of the scan cells are fed to the combinational circuit, for each vector shift, the inputs to the combinational logic change resulting in a large number of transitions during the test vector shifting process. Capture power is defined as power consumed when the normal model capture clock is applied so as to read the output response for a particular test vector that was scanned in.

In case of scan cells, a major power component is the unnecessary combinational transitions that occur during the test vectors shifts. This can be reduced by gating the outputs of the scan cells with the scan enable signal as shown in Figure 3.3. When the $SE = 0$, the circuit is in test mode and shifts the test vector sequence through the scan cells. Thus the outputs of the scan cells with respect the combinational logic is insignificant, and can be gated so as to avoid any unnecessary transitions. Additionally, we can also use the clock gating approach discussed in the previous section, to reduce the clock and flip flop cell power dissipation during test shifts (Figure 3.4).

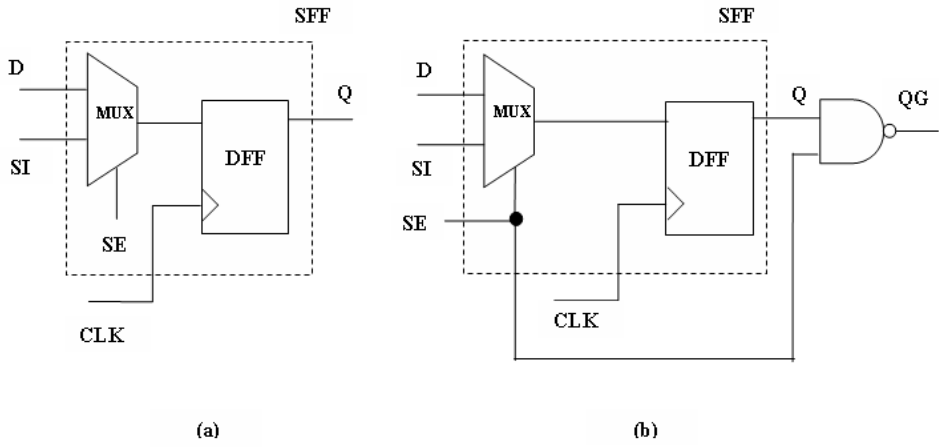

Figure 3.3: (a) A standard scan cell (SFF) and (b) A scan cell with Q output gated with a scan enable signal (SE) ($SE = 1$ is normal mode).

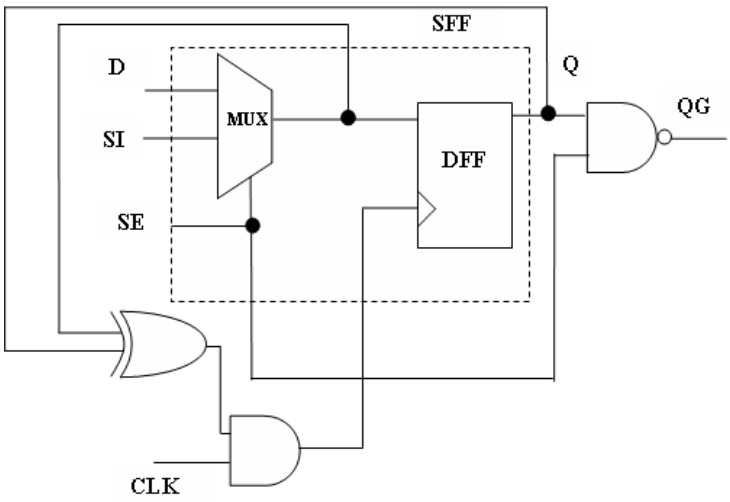

Figure 3.4: A scan cell with output Q gated with a scan enable Signal (SE) ($SE = 1$ is normal mode) and with clock gating.

Table 3.2: Power dissipation in ISCAS'89 benchmark circuit s5378 with scan cells of various types when the circuit is operated in normal mode.

Circuit name	No. of gates	Logic power (μW)	Glitch power (μW)	Dynamic power (μW)	Short Ckt. power (μW)	Leakage power (μW)	Clock power (μW)	Flip Flop power (μW)	Total power (μW)
$s5378_sff$	3137	81.76	19.5	101.28	13.92	0.13	220.26	751.7	1087.29
$s5378_sff_g$	3317	85.1	19.8	104.9	14.95	0.132	220.26	751.7	1091.94
$s5378_sff_g_clk$	3675	89.9	56.8	146.7	23.85	0.136	118.8	33.2	322.65

Table 3.3: Power dissipation in ISCAS'89 benchmark circuit s5378 with scan cells of various types when the circuit is being tested.

Circuit name	No. of gates	Logic power (μW)	Glitch power (μW)	Dynamic power (μW)	Short Ckt. power (μW)	Leakage power (μW)	Clock power (μW)	Flip Flop power (μW)	Total power (μW)
$s5378_sff$	3137	356.82	60.37	417.19	26.22	0.1459	220.26	848.53	1512.35
$s5378_sff_g$	3317	93.53	33.63	127.16	7.74	0.1504	220.26	850.69	1206.0
$s5378_sff_g_clk$	3675	146.78	241.89	388.67	61.9	0.1537	118.88	164.08	733.68

We have conducted experimental simulations on ISCAS'89 benchmark circuit s5378 with scan circuitry. We compare and observe the effect of the discussed low power scan cell design on the various power dissipation components. We explore the results for simulation in two modes (a) normal mode, Table 3.2 and (b) test mode, Table 3.3. The comparison is done between sequential circuit elements designed as a normal scan cell (Figure 3.3 (a)), a scan cell with gated Q output (Figure 3.3 (b)) and a Scan cell with gated Q output and with clock gating (Figure 3.4), respectively. In Tables 3.2 and 3.3, the three cases are shown as $s5378_sff$, $s5378_sff_g$ and $s5378_sff_g_clk$, respectively.

From Table 3.2, it is seen that in the normal mode, the circuit behaves similar to that of Table 3.1. The simulation is for 1000 random vectors with a clock period of 50 ns but with scan enable signal fixed for normal mode ($SE = 1$). As expected, both dynamic and leakage power components increase due to the added logic gates, while the flip flop cell power decreases due to clock gating. The effect of the gated low power scan cell, is not seen as the circuit is in the normal mode.

The test power results given in Table 3.3. We used test patterns generated by an ATPG program. Full scan design was done using Mentor Graphics FastScan and test patterns with a fault coverage of 98.87% were obtained. The combinational power is seen to decrease in the shift mode with the power gated scan cell design. The addition of clock gating decreases the clock and flip flop cell power during test shifts. It can be seen from the results, that the combinational power increases for the clock gating approach due to the added dissipation in the clock gating logic. Thus, circuits with large sequential depth and shallow combinational logic, the increase in combinational power due to extra transitions in the clock gating circuitry can be significant. However, in our example, the flip flop design potentially saves about 50% power during test.

3.10 Summary

In this chapter, we have discussed the techniques we have implemented for our gate level estimation tool. The tool is capable of separating and estimating the different power dissipation components. Some of the components we have focused our work on include dynamic,static,short circuit, clock and flip flop cell power and the significance of test power. In the coming chapters, our results will demonstrate that the techniques we have followed can accurately estimate the power dissipation while maintaining efficiency as compared to SPICE. The tool is also capable of separating and estimating the different power dissipation components.

Chapter 4

Power Estimation Results

4.1 Introduction

In this chapter we discuss the power analysis results we have obtained using our gate level power estimator tool. We will first do a walk through of the various steps involved in our program setup for conducting our power estimation. We then move on to our actual results, citing our underlying assumptions and technological constraints. We compare our results against the HSPICE [88] standard to verify the tool's accuracy and efficiency.

4.2 Experimental Procedure

Our analysis is done at the gate level design. The algorithm was implemented through a C/C++ program with the following steps.

- We read in the circuit as a simple netlist in a format called *rutmod* format. This basically a gate level description of the circuit giving information such as primary inputs, primary outputs, gate number, gate type and fan-in list. We have used a flattened netlist in our analysis.

- We also read in a vector file which contains the input vectors to the circuit. For power analysis processing we decided to use randomly generated vectors, generated using a random number generator written in C programming language. We also read in a technology file that has the various technological parameters like threshold voltage, oxide thickness, overlap capacitances etc. The user will have to enter certain

constraints like voltage supply used, vector period, and the kind of delay model to be used which enables the user to have a certain degree of control on the power analysis procedure.

- The power estimation tool outputs the following useful information after its analysis of the circuit:

    1. It reports circuit information like number of gates, total analysis period used and also the worst case delay for the particular series of vectors applied.
    2. The average power dissipation, including the total power, dynamic power (logic power and glitch power), short circuit power and leakage power. If the circuit is sequential in nature it will report the clock power and the flip flop cell power.
    3. It also gives other information like the maximum and minimum leakage causing vector, maximum and minimum power dissipation vector pair, the maximum number of glitches caused by a vector pair, etc.

4.3 Experimental Results

We first compare our estimation tool against the SPICE standard to verify the accuracy. Table 4.1 shows the results of our power analysis tool run on an simple inverter connected to a NAND and NOR load. The power results are for the inverter only and not for the whole circuit. An input signal with a rise/fall time of 1 ns was applied to the inverter input with a total analysis period of 100 ns. The circuit was implemented in two technologies: TSMC025 technology with a voltage supply of 2.5V and Berkeley predictive 90 nm technology [3] with a voltage supply of 1.0 V. The SPICE simulator gave us information like the total switching

Table 4.1: Comparison with SPICE using an INVERTER with a NAND and NOR load.

Techn. name	Input	SPICE Simulation			Our gate level estimator				
		Total power (μW)	Short ckt. current (μA)	Leakage power (pW)	Total power (μW)	Short ckt. current (μA)	Dynamic power (μW)	Short ckt. power (μW)	Leakage power (pW)
250 nm	0 - 1	1.08	17.10	24.01	1.031	16.32	0.6232	0.4077	22.65
	1 - 0	1.247	18.32	23.85	1.183	21.66	0.6424	0.5415	22.65
90 nm	0 - 1	0.098	6.37	6180	0.099	7.33	0.0258	0.0733	5890
	1 - 0	0.0978	6.16	6190	0.1436	11.25	0.0311	0.1125	5890

Table 4.2: 1-BIT ADDER Simulation

SPICE Simulation		Our gate level estimator				
Total power (μW)	CPU s	Total power (pW)	Dynamic power (μW)	Short circuit power (μW)	Leakage power (pW)	CPU s
4.09	180	2.58	1.23	1.35	451	5.4

power, leakage power and short circuit current, which are shown in columns 3, 4 and 5. We compare them to our estimation (columns 6 through 10), where we have successfully separated the switching energy into the dynamic and short circuit components. The results show that we have sufficient accuracy as compared to SPICE. We have also plotted the output voltage waveforms of the inverter as shown in Figure 4.1. Our estimate follows closely the SPICE waveforms. This helps in making accurate estimate of the output rise/fall times which in turn is important for accurate short circuit power estimation.

For further verification we ran a simulation on a 1-BIT adder circuit shown in Table 4.2. The 1-BIT adder was implemented in 0.25 micron technology. All possible input combinations were applied to the three inputs (A, B and Carryin), with a vector period of 100 ns and a rise time of 1 ns. The SPICE simulator took 180 ns while our estimator tool completed the simulation in 5.4 ns with reasonable accuracy.

We conducted our power analysis running our tool on the ISCAS benchmark circuits. The results shown in Table 4.3 are for a vector set of 1000 random vectors. Each input

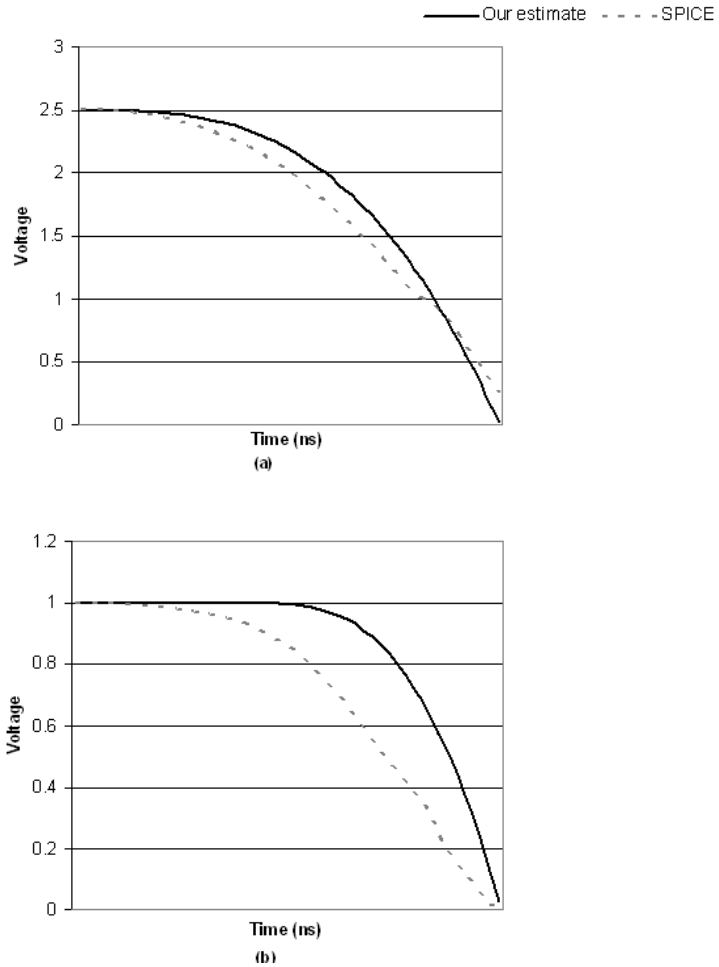

Figure 4.1: Output voltage waveforms for an inverter with a NAND and NOR load in (a) 0.25 micron (b) 90 nm technologies.

vector represents a pulse width of 100 ns with a rise/fall time of 1ns. The CPU times in Table 4.3 are for a Sun Sparc Ultra 10 with 4GB shared memory system

The tool can readily identify vectors that give the minimum, maximum and average power dissipation for each component. Figure 4.2 shows the progression of vectors for leakage power dissipation for the c880 benchmark circuit. Thus we are able to get a vector

Table 4.3: Average power dissipation for simulation of ISCAS Benchmark circuits using 1000 random vectors in 0.25 micron technology at a supply voltage of 2.5 volts.

Circuit name	No. of gates	Logic power (μW)	Glitch power (μW)	Dynamic power (μW)	Short Ckt. power (μW)	Leakage power (μW)	Total power (μW)	CPU s
c880	383	38.26	24.89	63.16	49.8	0.0149	112.99	195.84
c1355	546	70.39	37.1	107.49	81.75	0.0180	189.26	252.77
c1908	880	125.5	101.06	226.57	52.04	0.0285	278.64	570.7
c2670	1193	160.87	177.86	338.74	116.88	0.0477	455.68	1028.7
c3540	1669	198.01	250.77	448.78	125.83	0.0651	574.69	1347.6
c5315	2307	384.65	391.44	776.09	238.12	0.0950	1014.31	1921.1
c6288	2416	298.88	3841.54	4140.41	146.68	0.08277	4288.05	7564.8
c7552	3512	533.80	659.63	1193.44	230.32	0.133	1423.91	3047.5

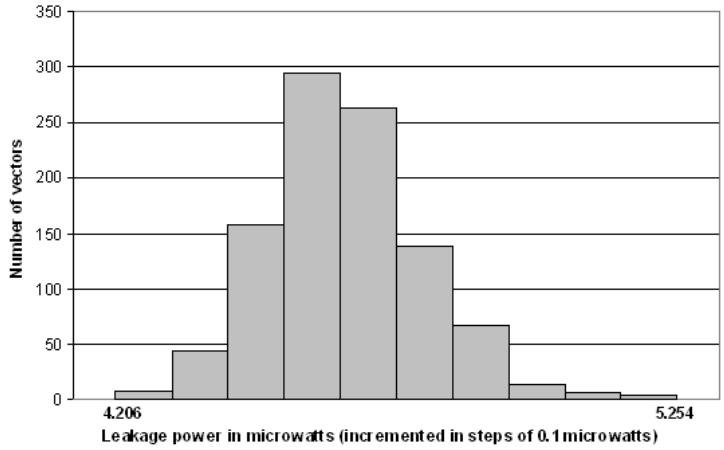

Figure 4.2: Leakage power of c880 in 90nm technology for 1000 random vectors.

set that can effectively give the least leakage possible for this circuit. This vector set is useful as it can be used to reduce leakage in circuits in standby mode.

We conducted an experiment to study the short circuit power dissipation. We analyzed an inverter chain circuit with 6 inverters, with the last inverter being used as an output load. In the first case, we sized the inverters equally with the least size available for the 0.25 micron technology. In the second case, we increased the sizes by progressively doubling the sizes and in the third scenario we decreased the sizes in the same manner. The input vector signal

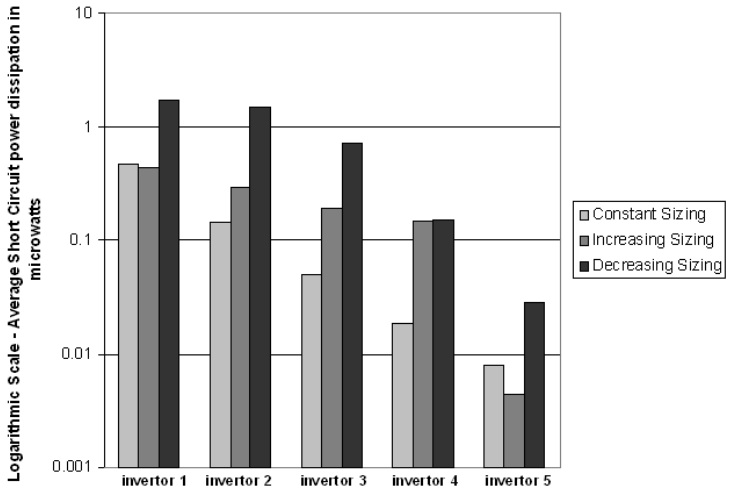

Figure 4.3: Effect of gate sizing on short circuit power dissipation.

given was a 0→1→0 with a pulse width of 100 ns and a rise time of 1 ns .The effect of the rise or fall times was nullified as the increase or decrease in the resistance was compensated by the decrease or increase in the load capacitances, respectively. Thus the effect of the load capacitances became the determining factor for the short circuit dissipation. The results shown in Figure 4.3 are the average short circuit dissipation for each inverter for each of our sizing scenarios. As seen by our results, progressively decreasing the sizes gave the highest overall short circuit power which can be accounted for by the decreasing output load capacitances coupled with the high drivability of the inverters. From this analysis we can see that it may be possible and useful to optimize the gate sizes of gates in such a way as to decrease the overall short circuit power.

4.4 Summary

In this chapter we have described our experimental results using our gate level power estimator tool. We have compared our results against the SPICE standard and found our

estimation to be of reasonable accuracy. We have obtained results for the ISCAS benchmark circuits and presented experimental analysis of few circuits to show the usefulness of such a tool.

Chapter 5

Bounded Delay and Dynamic Power Estimation

5.1 Introduction

In this chapter, we discuss bounded delay principles and their usefulness in handling uncertainties in gate delay values due to process variation. We define and then discuss *ambiguity regions*, putting forward several theorems to determine them accurately. We give a novel technique for using the bounded delay information to estimate dynamic power under variation in delay values.

5.2 Background and Definitions

To deal with the variability, significant advances have been made in logic simulation, timing analysis and delay testing areas with the use of the *bounded delay* [82] model. Both statistical and bounded delay models have been used [7, 52, 87]. In the bounded delay model, each gate is assigned the lower and upper bounds for delays, also called the *min-max* delay specification [12, 13, 18, 19, 37, 54, 80, 82].

To define and represent signals in bounded delay models, we use the term *ambiguity region* as a region of signal uncertainty where we cannot deterministically tell when the signal transitions within that region. The ambiguity region is depicted in Figure 5.1 and is described by:

- EA is the earliest arrival time for a signal.
- LS is the latest stabilization time for a signal.

Figure 5.1: Ambiguity regions of a rising signal (a) and a steady state signal (b) respectively.

- IV is the initial value of a signal.
- FV is the final value of a signal.

In the following sections we will describe our theorems to accurately determine the ambiguity regions, that is, EA and LS values, from the bounded gate delays.

5.3 Signal Transition Analysis of Bounded Delay Gates

In this work we aim to estimate the number of transitions that the output of a gate makes for each input vector pair. It is assumed that the steady-state values, which can be predetermined by zero-delay logic simulation, are known. In the next sections we will describe our techniques to statically determine the ambiguity intervals (transition periods) and the bounds on the number of transitions that the gate output would make.

5.3.1 Ambiguity intervals

Our analysis follows the events that occur at gate outputs in a circuit after a change, typically on application of a new vector, occurs at primary inputs. We determine the ambiguity (transient) interval for the signal at the output of a gate from,

1. Bounded delays of the gate.
2. Steady-state signal values, that is, *initial value (IV)* and *final value (FV)*.

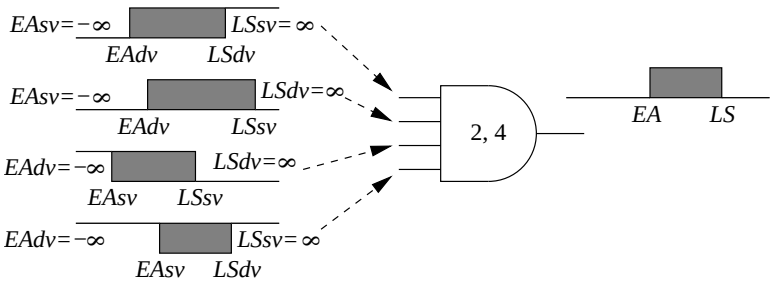

Figure 5.2: A four-input AND gate with delay bounds (2, 4). Shaded regions are ambiguity intervals.

We will assume that the primary inputs change at deterministic times, synchronized with a clock. In general, however, input change ambiguities can be specified and treated in a similar way as described here.

All times defined here are with reference to the start of the clock period. Because all gates are assumed to have delays within their respective specified (min, max) bounds, a typical gate's inputs go through transient intervals before settling to some final values (FV). In the bounded delay formulation we do not precisely know how many transitions each signal makes. As explained earlier the ambiguity region is defined by the earliest arrival time of a signal (EA), and the latest stabilization time of a signal (LS). We derive certain variables from (EA, LS) which are dependent on (IV, FV) and on the particular type of gate it effects. For clarity, we will use the example of an AND gate with four inputs as shown in Figure 5.2. Borrowing from the literature [12], we define:

- $EAdv$ is the earliest arrival time of a signal that causes the input of the gate to change from controlling value (e.g., 0 for AND gate) to non-controlling value (1 for AND gate).

- $LSdv$ is the latest stabilization time of an input signal changing from controlling value to non-controlling value.

- $EAsv$ is the earliest arrival time of an input signal changing from a non - controlling value to a controlling value.

- $LSsv$ is the latest stabilization time for an input signal changing from a non - controlling value to a controlling value.

It should be noted that for $EA = \infty$ and $LS = -\infty$, the output is defined as having no ambiguity region or is in steady state condition. The following theorem determines the output ambiguity interval.

Theorem 1: The ambiguity interval (EA, LS) for the output signal of a logic gate is determined from the ambiguity intervals of input signals, their pre-transition and post-transition steady-state values, and minimum and maximum gate delays, as follows. Considering all inputs i of the gate, we define:

$$E1 = maximum\{EAdv(i)\} \tag{5.1}$$

$$E2 = minimum\{EAsv(i)\} \tag{5.2}$$

$$L1 = minimum\{LSsv(i)\} \tag{5.3}$$

$$L2 = maximum\{LSdv(i)\} \tag{5.4}$$

$$EA' = maximum\{E1, E2\} \tag{5.5}$$

$$LS' = minimum\{L1, L2\} \tag{5.6}$$

Then

$$(EA, LS) = \begin{cases} \{EA' + mindel, LS' + maxdel\} \\ \qquad\qquad \text{if } (LS' - EA') \geq maxdel \\ \\ \{\infty, -\infty\} \qquad \text{if } (LS' - EA') < mindel \end{cases}$$

where the inertial delay of the gate is bounded as $(mindel,\ maxdel)$. ■

In general, the output of a gate can have multiple ambiguity regions separated by deterministic signal values as we will demonstrate in the next section. In that case, each ambiguity region as well as each deterministic interval will be affected by the inertial filtering caused by *mindel*. For simplicity, Theorem 1 takes a pessimistic view by combining all possible ambiguity regions into one.

Once the ambiguity interval (EA, LS) is determined according to Theorem 1, the steady-state values allow a straightforward conversion to the detailed signal specification. For the example of Figure 5.2, at the output of the AND gate, $EA = EAdv$ and $LS = LSsv$. Also notice when IV takes a dominant (non-dominant) value, $EAsv$ $(EAdv) = -\infty$. Similarly, when FV takes dominant (non-dominant) value, $LSdv$ $(LSsv) = \infty$. $(EA, LS) = \{\infty, -\infty\}$ means that the output pulse is completely suppressed due to gate inertia.

5.3.2 Multiple ambiguity regions

In certain cases, multiple ambiguity regions may rise in outputs that are separated by regions of deterministic signal states. Unlike in timing scenarios where only the outside bounds of the ambiguity region are important, in case of power estimation, we need to

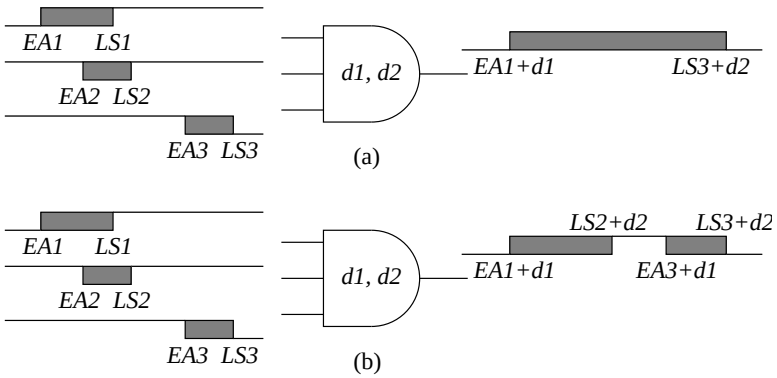

Figure 5.3: A three-input AND gate depicting multiple ambiguity intervals.

address this issue. Without considering the multiple regions within an ambiguity period, we may overestimate or underestimate the power. For this we follow a simple procedure. We first arrange all input (EA, LS) values, in order of their temporal occurrence. After we calculate the bounds, (EA', LS') from *Theorem 1*, we examine the EA and LS values of the inputs within these bounds. If any LS occurs immediately earlier than an EA value, then a multiple ambiguity region occurs and we propagate this value to the output, only if any two consecutive bound values are spaced at least the gate inertial delay apart. The example in Figure 5.3 depicts this. In (a), when we do not consider multiple intervals, we get a final output ambiguity region of $(EA1 + d1, LS3 + d2)$. However in (b), we can see the difference. Since $LS2$ occurs earlier than $EA3$, clearly this would pass through the AND gate if the bound difference is greater than inertial delay d. Thus the corrected output ambiguity region will have multiple intervals of $(EA1 + d1, LS2 + d2, EA3 + d1, LS3 + d2)$. The algorithm to implement this is given below:

1: **for all** gates (G) with multiambiguity inputs **do**

2: $EAmin = EA(G) - mindel(G)$

3: $LSmax = LS(G) - maxdel(G)$

```
4:  for all gates (g) in fanin(G) do
5:    if EA(g) ≠ ∞ and LS(g) ≠ −∞ then
6:      if (EA(g), LS(g)) > EAmin and (EA(g), LS(g)) < LSmax then
7:        addtoBoundsList(EA(g), LS(g))
8:      end if
9:    end if
10: end for
11: for all (EA,LS) values t(i) in addtoBoundsList do
12:   sort t(i) in increasing order
13: end for
14: for all (EA,LS) values t(i) in addtoBoundsList do
15:   if t(i) is of type 'LS' and t(i + 1) is of type 'EA' then
16:     if (t(i + 1) − t(i)) ≥ maxdel(G) then
17:       addtoGateBoundsList(t(i) + maxdel(G), t(i + 1) + mindel(G))
18:     end if
19:   end if
20: end for
21: end for
```

5.4 Problem Depiction

A simple estimate for output transitions would be to take the sum of all fan-in transitions, assuming they will all propagate through the gate. However, such a prediction is too pessimistic and we will show that this would give a very high bound for the maximum

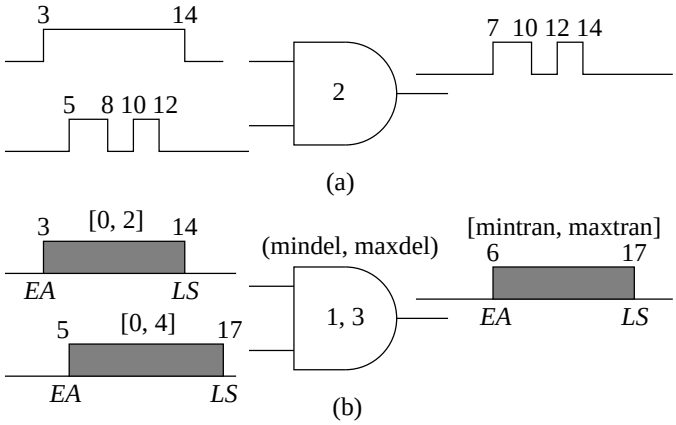

Figure 5.4: Two-input AND gate transitions.

power. Similarly using only the steady state values to predict the minimum power, i.e., zero transition if $IV = FV$ and one transition if $IV \neq FV$, will give a very low bound on minimum power. Theoretically, though these bounds are correct, our aim is to get more precise bounds with a fast and efficient method.

The two-input AND gate example in Figure 5.4 depicts the problem we aim to solve. In (a) the gate has a deterministic delay (2) and thus we know, for the given fan-in signals, that the output will have 4 transitions. However in (b), we have two fan-in ambiguity intervals which are $(EA, LS) = (3, 14)$ and $(5, 17)$, respectively. Also, the gate has delay bounds (1,3). Let us assume from the analysis of previous gates we know that minimum and maximum numbers ($mintran$, $maxtran$) of transitions for two fan-ins are, respectively, (0,2) and (0,4). With this information, we aim to obtain at the gate output a deterministic maximum bound for transitions $maxtran$ and a minimum bound for transitions $mintran$. The next sections explain our techniques for determining these.

5.5 Maximum Number of Transitions

Assuming that primary inputs are glitch-free, the number of transitions (0 or 1) there for a vector-pair is known. Using the algorithms of this and the next subsections iteratively, we can determine the bounds on the numbers of transitions for all signals.

Consider a gate. Given data consists of ambiguity intervals and the minimum and maximum transitions, *mintran* and *maxtran*, respectively, for fan-ins. We will estimate the value of *maxtran* at the output. We consider two things: (1) cause - an output transition must be caused by an input transition, and (2) filtering - gate inertia can filter out transitions that are closer to each other than the gate delay. In the absence of detailed information, we assume that the transitions at a fan-in are evenly spaced within its ambiguity interval.

Agrawal et al. [5] have derived two upper bounds for the number of events possible at the output of a gate. However, in their derivation, neither ambiguity regions nor the initial (IV) and final (FV) output values have been considered. We improve upon those bounds by considering these factors in the following theorem.

Theorem 2: The maximum number of transitions is defined as the minimum of the two upper bounds:

$$maxtran \quad = \quad minimum(Nd, N) \tag{5.7}$$

where Nd is the maximum number of transitions permitted by the gate inertial delay and N is the sum of all transitions present at the gate input.

Proof: We derive the two upper bounds for maximum transition and take the lower of those as a tighter upper bound. This analysis is an improvement over a previously reported result [5].

First upper bound (Nd): We calculate the maximum number of transitions that can be accommodated in the ambiguity interval given by the gate delay bounds and the (IV, FV) output values.

We consider the filtering of glitches by gate inertia. Note that most transitions can be accommodated if they are evenly spaced over the output ambiguity interval with a spacing equal to or greater than the inertial delay. We consider the following cases:

1. If the output has a static hazard, then we allow an even number of transitions determined by $tpd \times (2n-1) \leq LS - EA$, and the number of transitions is given by $2n$, where tpd is the gate delay given by minimum delay bounds, n is the number of hazards that can possibly be accommodated in the ambiguity interval, LS is the latest stabilization time and EA is the earliest arrival time for the output signal, as given by Theorem 1.

2. Similarly, for an output signal with a dynamic hazard we would get an odd number of transitions determined by $tpd \times 2n \leq LS - EA$, and the number of transitions is given by $2n-1$.

Second upper bound (N): We modify the sum ($Nsum$) of the input transitions as:

$$N = Nsum - k \tag{5.8}$$

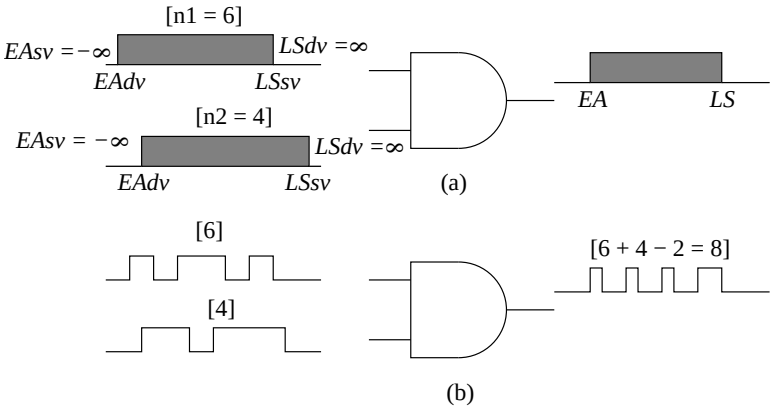

Figure 5.5: Effect of modification factor k on the second upper bound.

where $k = 0$, 1, or 2 for a 2-input gate and is determined by the ambiguity regions and $(IV,\ FV)$ values of inputs. The procedure is explained by the example of Figure 5.5. In (a), for the given input transitions $n1 = 6$ and $n2 = 4$, the output cannot have the total sum of input transitions. This is because, when we consider the sum, in the case of two signals going from a controlling value to a non-controlling value, only one of the two transitions should be counted. Thus. we see that a correction factor k is required, which would give us a $maxtran = 8$ for the example in (a). The example in (b) is a possible deterministic signal representation of the same. ■

An application of the two upper bounds is shown in Figure 5.6. In (a) the gate is assumed to have zero delay. Thus from the above principles we get the maximum number of transitions from the first upper bound, $Nd = \infty$ and the second upper bound gives us $N = 8$. The minimum of the two gives us the final $maxtran = 8$ value. However when we consider the gate delay bounds (3,5) as in (b) we find that 8 transitions can never occur within the output ambiguity bounds of (6,23). Using the above principles we get $Nd = 6$ and $N = 8$. In fact, the maximum possible $maxtran$ is only 6.

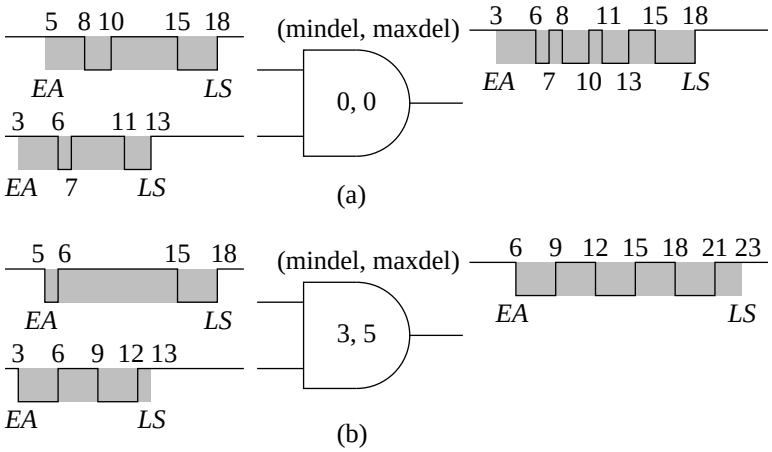

Figure 5.6: Filtering of transitions in a two-input AND gate.

5.6 Minimum Number of Transitions

A rather pessimistic lower bound on minimum number of transitions, *mintran*, can be found from the steady-state values at the gate output. This bound is 0 or 1 depending upon whether the output values before and after transients, i.e., IV and FV, are same or different. This has been used as the condition for minimum glitch power design [5]. When there are split ambiguity regions, we can obtain a tighter lower bound.

Theorem 3: The minimum number of transitions is higher of the two lower bounds:

$$mintran \quad = \quad maximum(Ns, Ndet) \tag{5.9}$$

where *Ns* is the number of transitions required by the steady-state signal values and *Ndet* is that needed to produce deterministic signal values separating any non-overlapping ambiguity intervals.

Proof: *First lower bound* (Ns): This is obtained from the steady-state values, without considering any details of the ambiguity region. Logic changes 0→0, 1→1 need not have any

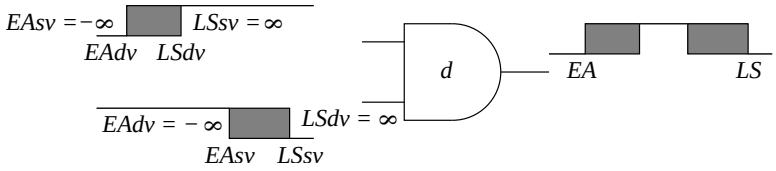

Figure 5.7: Estimating lower bound on output transitions of a 2-input AND gate.

transition and 0→1, 1→0 must have at least one transition. Provably, $Ns = 1$ if $IV \neq FV$, and $Ns = 0$ if $IV = FV$ [5].

Second lower bound ($Ndet$): The number of definite transitions that can occur in the output ambiguity region is the number of deterministic signal changes that occur within the ambiguity region such that signal changes are spaced at time intervals greater than or equal to the inertial delay of the gate.

The effect of the second lower bound can be seen in the example of Figure 5.7. There are at least two essential signal changes that must occur within the output ambiguity region. Thus, there will always be a hazard in the output as long as:

$$(EAsv - LSdv) \quad \geq \quad maxdel \tag{5.10}$$

where *maxdel* is the maximum delay of the gate producing the transient. ■

In this case *mintran* is not zero, as given by the steady state first lower bound, but is 2. Detailed analysis of split ambiguity regions is possible with the help of *transient output functions* (TOF) [58] or *timed Boolean functions* (TBF) [53, 59]. In general, the value of $Ndet$ can be higher than 2 depending on how many ambiguity and deterministic regions are produced and their widths with respect to the gate delay d. This can be easily estimated from the technique used to determine the multi ambiguity regions in subsection 5.3.2.

5.7 Dynamic Power Estimation

The inputs to analysis are a gate-level combinational circuit netlist, in which each gate has two delay bounds (mindel, maxdel) and a node capacitance, and a set of vectors. Capacitances may be extracted from layout or estimated using a wire-load model. Nominal delays for all gate types are precharacterized using SPICE-simulator, which also determines and saves the input-dependent leakage current data in a library. Manufacturing technology data on percentage process variation ($\Delta\%$) is used to determine the bounded delay specification, *maxdel, mindel* = *nominal delay* $\pm\Delta\%$.

Dynamic power estimation is a three pass procedure for each input vector, performed in level-order for all gates:

1. The first pass is zero-delay logic simulation that determines the initial and final values, IV and FV, for all signals.

2. The second pass determines the earliest arrival (EA) and latest stabilization (LS) times according to Theorem 1 for all signals using the precalculated IV and FV, and the gate delay bounds.

3. The third pass determines the upper and lower bound, *maxtran* and *mintran*, for all gates according to Sections 5.5 and 5.6. For each primary input, we assume *maxtran* = *mintran* = 0, if the present and previous vectors have the same value, or = 1, if they have different values.

5.8 Summary

In this chapter, we have discussed our bounded delay analysis method for estimating the dynamic power under uncertainties in gate delays. The uncertainties in delays can be caused due to process variations and other parameter changes. We discuss principles like ambiguity regions and put forward our theorems for estimating the delay bounds as well as the maximum and minimum transitions possible under delay uncertainties. In our results section we will show that our technique is far faster than traditional Monte Carlo simulations without any loss in accuracy.

Chapter 6

Bounded Delay Analysis Results

6.1 Introduction

In this chapter we discuss the various experimental results we obtained using our bounded delay analysis algorithm for dynamic power estimation. We will first talk about the various steps involved in our program setup for conducting our experiments. We then move on to our actual results citing our underlying assumptions and technological constraints.

6.2 Experimental Procedure

Our analysis is done at the gate level design. The algorithm was implemented through a C program with the following steps.

- We read in the circuit as a simple netlist in a format called *rutmod* (Rutgers modeling language) format as described in Section 4.
- We also read in a vector file which contains the input vectors to the circuit. A capacitance library file is used in which each gate number and the corresponding fan-out load capacitance is listed. This is pre-computed via SPICE characterization, which is specific to a technology file. Last we read in a delay file which has the gate number, the nominal delay (computed by a wire load delay model), the minimum bounded gate delay and the maximum bounded gate delay.
- The outputs of the program are two result files. The first contains the following: gate number, initial output value, final output value, minimum number of transitions

Table 6.1: Per vector energy consumption in picojoule in benchmark circuits for 1000 random vectors by Monte Carlo simulation of 1000 sample circuits and bounded delay analysis.

Circuit name	Monte Carlo simulation *picojoule per vector*				Bounded delay analysis *picojoule per vector*			
	Minimum	Maximum	Average	CPU s	Minimum	Maximum	Average	CPU s
c880	1.086	10.847	4.340	298.26	1.080	11.140	4.240	0.34
c1355	3.606	13.577	7.310	423.69	3.600	20.150	10.928	0.59
c1908	4.870	29.470	15.580	840.85	4.590	57.050	17.750	0.69
c2670	8.470	51.190	24.390	1452.24	8.390	59.010	23.200	1.09
c3540	6.036	66.660	30.770	1810.18	5.970	96.180	35.100	1.39
c5315	29.810	91.100	56.41	3435.53	23.030	113.200	55.610	2.14
c6288	45.360	194.860	129.700	20944.53	11.840	406.340	153.710	2.60
c7552	35.050	146.120	82.790	5834.87	29.470	196.310	82.180	3.34

and maximum number of transitions. The second gives the maximum, minimum and mean power consumption of the circuit for each vector pair.

6.3 Benchmark Results

Our first results consist of the power analysis of ISCAS85 benchmark circuits for 1000 random vectors. The circuits were implemented using the TSMC025 2.5V CMOS library. Process variation can be modeled by assuming 15% intra-die and 5% inter-die variation [47]. For illustrative purposes, a standard size gate delay of 10ps and wire-load delay model were used to determine the nominal gate delays from which bounds (mindel, maxdel) were obtained by assuming a $\pm 20\%$ variation. It should be noted that any kind of variation modeled is suitable for our method. Node capacitances for the circuits as mentioned ealier were obtained from the SPICE modeling files.

Table 6.1 gives two sets of data. The first set (columns 2-5) is from a Monte Carlo event-driven simulation. These results are for 1000 circuit samples. Each gate in a sample circuit was assigned a delay using a random number uniformly distributed in its (mindel, maxdel) range. From the simulation of 1000 random vectors, we have listed the energy

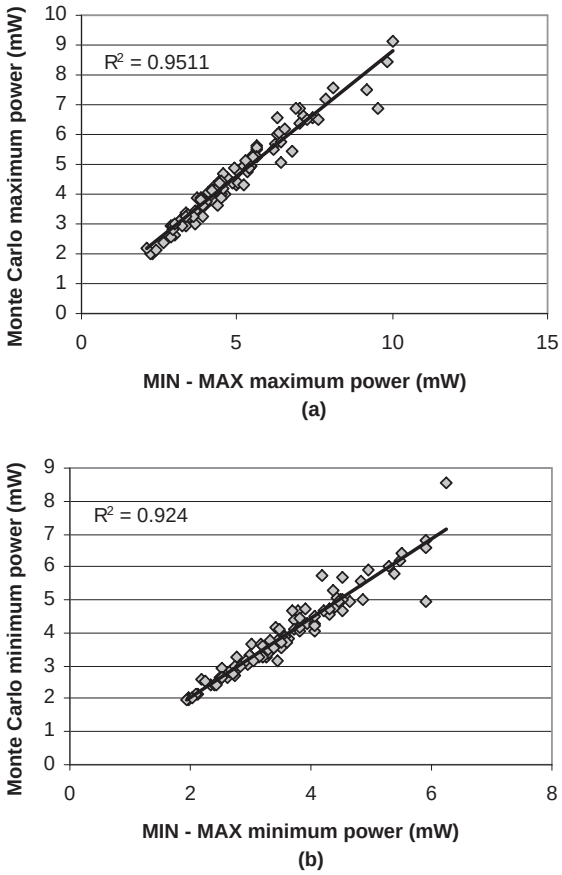

Figure 6.1: Monte Carlo simulation versus bounded delay analysis for c880. Each point represents one vector-pair. One hundred sample circuits with *nominal* $\pm$ 20% delay variation were simulated and for each vector-pair (a) maximum and (b) minimum power was determined.

consumption in picojoule (pJ) for two vectors consuming the least and the most energy, and the average for all 1000 vectors. We observe that the bounded delay analysis always gives lower minimum energy and higher maximum energy. This is expected. As we simulate more vectors, the event-driven numbers drop on the minimum side and increase on the maximum side. Besides, to take the variability into account, the event-driven simulator will have to be used in a Monte Carlo experiment mode, which will take much more computing resources.

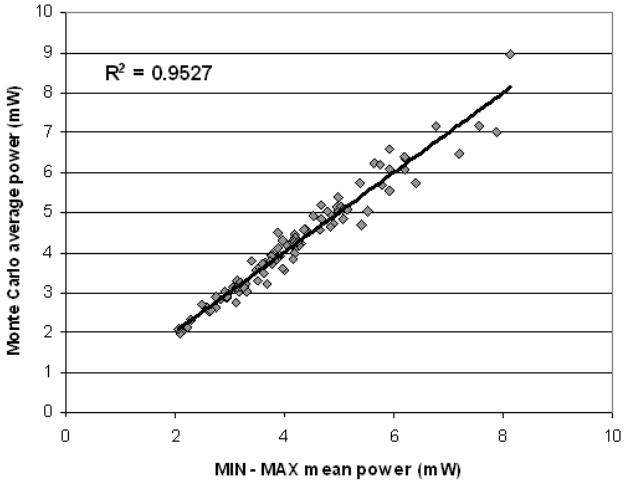

Figure 6.2: Monte Carlo simulation versus bounded delay analysis for c880. Regression graph for average power.

The CPU times in Table 6.1 are simulation runs on a UNIX system using an Intel Duo Core processor with 2GB RAM.

We next conducted a Monte Carlo analysis using the event-driven simulator. As described before, 100 samples were generated for c880. Each sample circuit was simulated by the event-driven simulator for the same set of 100 random vectors. For each vector-pair, we obtained the minimum, average and maximum power. The maximum and minimum numbers are shown in Figure 6.1 against the corresponding vector-pair results from the bounded delay analysis. Here the power was calculated using a vector application period of 1ns. R^2 shown on the regression graphs is the *coefficient of determination* from Microsoft Excel, whose ideal fit value is 1.0. For a similar regression graph for average power shown in Figure 6.2, $R^2 = 0.9527$, showing a reasonable accuracy. Note that the statistical distribution in Figure 1.1 is skewed. Our average power is the simple arithmetic mean of the

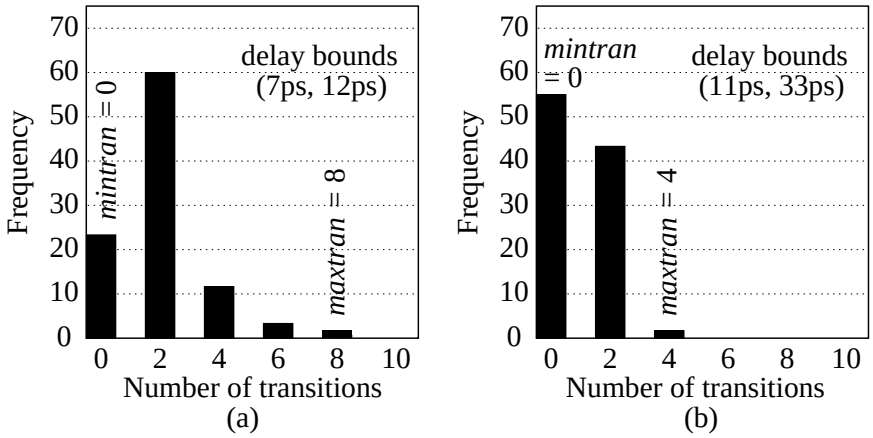

Figure 6.3: Transition statistics for high-activity gate 1407 in c2670 for a random vector-pair. Bounded delay analysis: (a) delay bounds (7ps, 12ps), $mintran = 0$, $maxtran = 8$, (b) delay bounds (11ps, 33ps), $mintran = 0$, $maxtran = 4$. Histograms were obtained by Monte Carlo simulation.

minimum and maximum and assumes a symmetric distribution; an improvement may be possible.

Figure 6.3 shows the transition statistics for gate 1407 in c2670. This is a high activity gate, which made up to 8 transitions on some vector-pairs. The left histogram (a) shows the number of transitions on this gate for one vector-pair applied to 100 sample circuits. The delay bounds of the gate were (7ps, 12ps). So, in each sample, its delay was randomly selected from this range. The transitions on the gate range between 0 and 8. Bounded delay analysis gave $mintran = 0$ and $maxtran = 8$. Leaving all other gate delays as before, when the delay bounds of 1407 were changed to (11ps, 33ps), the analysis computed $mintran = 0$ and $maxtran = 4$. The corresponding histogram from Monte Carlo simulation is shown in Figure 6.3(b).

Figure 6.4 depicts the maximum power distribution obtained from simulation of two ISCAS '85 benchmark circuits, c880 and c5315. It shows the comparative histograms of the Monte Carlo simulations (depicted by the light coloured bars) and our bounded delay

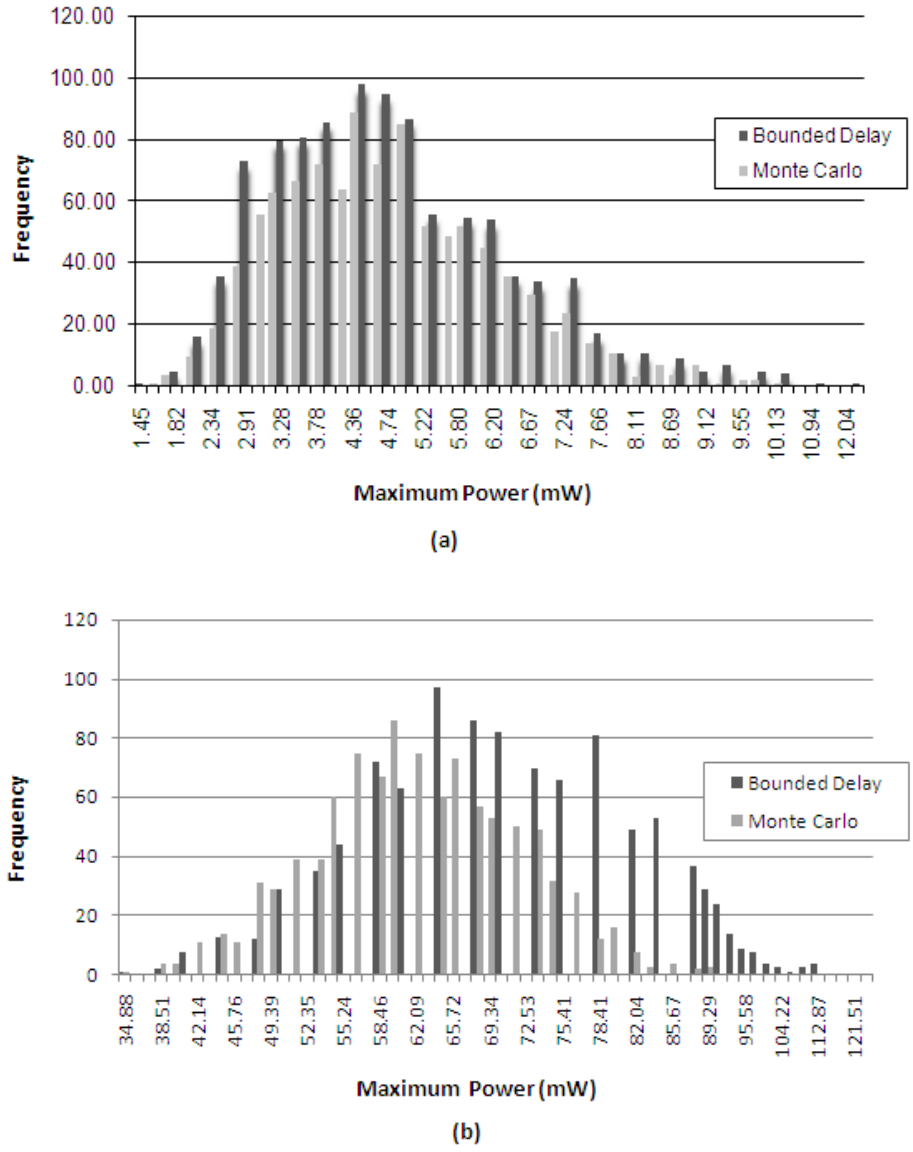

Figure 6.4: A comparison of the maximum power distribution for a vector-pair obtained by bounded delay analysis and Monte Carlo simulation for ISCAS '85 benchmark circuits (a) c880 and (b) c5315. The maximum power values are for 1000 random vector pairs. The Monte Carlo simulation used 1000 circuit samples with random delays to find the maximum power for each vector pair.

analysis technique (shown by darker bars). The maximum power on the horizontal axis is the maximum power dissipation of a particular vector pair. The gray bars give the maximum value obtained by Monte Carlo simulation of 1000 delay samples with a delay variation of 20% for each of the 1000 vector pairs. As can be seen the bounded delay estimate closely follows Monte Carlo analysis. It should also be noted that the Monte Carlo maximum power distribution would be closer to the bounded delay result as we increase the number of delay samples.

The average of maximum power tends to remain unchanged with increasing number of vector pairs. The peak maximum power also tends to converge for a large number of vector simulations. One can employ the statistical method of *extreme order statistic* to determine the peak maximum power from a random vector set as discussed in the literature [6, 72, 98]. Qiu *et al.* [72, 98] propose a technique through which they estimate maximum power with a confidence level of 90% for 5% error. They simulate only 2500 vector pairs. Here, a vector sample of size n is randomly simulated m times. The maximum power is obtained for each m simulation which tends to follow a generalized weibull distribution. The problem is equivalent to calculating the location parameter of the weibull distribution from random samples. The authors do so by using a maximum likelihood estimator which converges to a normal distribution.

6.4 Summary

In this chapter we have described experimental results on the ISCAS benchmark circuits using our bounded delay analysis algorithm. We have verified both the accuracy and speed

of our approach. The new technique is shown to be far more efficient than the traditional Monte Carlo simulation and can serve as a useful alternative.

Chapter 7

Conclusion

Low power design has become an important issue in solving the four-fold design problem of area, performance, power and testability. Improved CAD systems and other low power estimation and optimization tools are necessary to give the designer adequate support. From our own observations, a power estimation tool can give useful information on the impact of clock power consumption, short circuit dependencies and the effects of variations in power estimation. In this thesis we tried to study the effects of all these factors on power consumption, providing our own implementations for estimating the various components. We have successfully implemented a gate level power estimator tool that can separate different power dissipation components and provide the designer information about their effects on the total power consumption. The tool has been applied to combinational benchmark circuits and experimental results were validated against SPICE.

We also developed an efficient power estimation method with consideration of process variations. Our present target is dynamic power. We have used the min-max (bounded) delay model and developed new algorithms to determine bounds on gate transitions. This analysis has a linear-time complexity in number of gates and is an efficient alternative to the Monte Carlo analysis. Presently, we can include leakage based on signal states that are obtained from inherent zero-delay simulation in this method. Our expectation for the future is to consider process variation in leakage as well. Besides, node capacitances that are considered fixed here can also have process-dependent variation. We hope to investigate that in the future.

For digital CMOS circuit technologies, process variation and leakage power will continue to present significant design challenges. In low-power design, a combined optimization over multiple power components, such as leakage reduction by dual-threshold design and glitch reduction by path delay balancing, has been attempted [55, 56, 57]. Components of power are not independent and reduction of one component may affect the other. An analysis tool of the type discussed here is useful. Similarly, process variation can wipe out the benefits of a power optimization technique unless variation is considered in the design [40]. It is expected that the bounded delay methods discussed in the present research will be adopted in power optimization procedures.

The analysis techniques discussed in the present work are simulation based. When a selected vector set defines the application domain, it can be used for vector-specific power optimization [43]. A simulation-based power analysis method is then a very effective evaluation tool. In many cases, however, the defining vector set may be either too large or too difficult to find. In those cases, static or vector-less power analysis may be sufficient. Although vector independent approaches are, in general, less accurate they are significantly more efficient than the dynamic (simulation-based) analysis. The bounded delay power analysis algorithms of the present work can be modified for static analysis.

Bibliography

[1] http://www.mosis.com/.

[2] http://www.eecs.umich.edu/~jhayes/iscas.restore/benchmark.html.

[3] http://www.eas.asu.edu/~ptm/.

[4] V. D. Agrawal, "Low-Power Design by Hazard Filtering," in *Proc. Tenth International Conf. on VLSI Design*, Jan. 1997, pp. 193–197.

[5] 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. Twelfth International Conference on VLSI Design*, Jan. 1999, pp. 434–439.

[6] V. Bartkute and L. Sakalauskas, "Three Parameter Estimation of the Weibull Distribution by Order Statistics," in C. H. Skiadas, editor, *Recent Advances in Stochastic Modeling and Data Analysis*, pp. 91–100, World Scientific, 2007.

[7] J. W. Bierbauer, J. A. Eiseman, F. A. Fazal, and J. J. Kulikowski, "System Simulation With MIDAS," *AT&T Tech. J.*, vol. 70, no. 1, pp. 36–51, Jan. 1991.

[8] L. Bisdounis, S. Nikolaidis, and O. Loufopavlou, "Propagation Delay and Short-Circuit Power Dissipation Modeling of the CMOS Inverter," *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol. 45, no. 3, pp. 259–270, Mar. 1998.

[9] S. Bobba and I. N. Hajj, "Maximum Leakage Power Estimation for CMOS Circuits," in *Proc. 15th International Conf. on VLSI Design and 7th Asia and South Pacific Design Automation Conf.*, Jan. 1999, pp. 116–124.

[10] A. Bogiolo, L. Benini, and B. Riccò, "Power Estimation of Cell-Based CMOS Circuits," in *Proc. Design Automation Conf.*, 1996, pp. 433–438.

[11] A. Boliolo, L. Benini, G. de Micheli, and B. Ricco, "Gate-Level Power and Current Simulation of CMOS Integrated Circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 5, pp. 473–488, Dec. 1997.

[12] S. Bose and V. D. Agrawal, "Delay Test Quality Evaluation Using Bounded Gate Delays," in *Proc. 25th IEEE VLSI Test Symp.*, May 2007, pp. 23–28.

[13] S. Bose, H. Grimes, and V. D. Agrawal, "Delay Fault Simulation With Bounded Gate Delay Model," in *Proc. of International Test Conf.*, 2007, pp. 23–28.

[14] R. E. Bryant, "MOSSIM: A Switch-level Simulator for MOS VLSI," in *Proc. 18th Design Automation Conference*, July 1981, pp. 786–790.

[15] R. Burch, F. Najm, P. Yang, and D. Hocevar, "Pattern-Independent Current Estimation for Reliability Analysis of CMOS Circuits," in *Proc. 25th ACM/IEEE Design Automation Conf.*, June 1988, pp. 294–299.

[16] R. Burch, F. Najm, P. Yang, and T. Trick, "McPOWER: A Monte Carlo Approach to Power Estimation," *Proc. IEEE/ACM International Conference on Computer-Aided Design*, pp. 90–97, Nov 1992.

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

[18] S. Chakraborty and D. L. Dill, "More Accurate Polynomial-Time Min-Max Timing Simulation," in *Proc. Third International Symp. Advanced Research in Asynchronous Circuits and Systems*, Apr. 1997, pp. 112–123.

[19] S. Chakraborty, D. L. Dill, and K. Y. Yun, "Min-Max Timing Analysis and an Application to Asynchronous Circuits," *Proc. of IEEE*, vol. 87, no. 2, pp. 332–346, Feb. 1999.

[20] S. Chandra, K. Lahiri, A. Raghunathan, and S. Dey, "Considering Process Variations During System-level Power Analysis," in *Proc. International Symposium on Low Power Electronics and Design*, 2006.

[21] A. P. Chandrakasan and R. W. Brodersen, *Low power Digital CMOS Design*. Springer, 1995.

[22] H. Chang and S. S. Sapatnekar, "Full-Chip Analysis of Leakage Power Under Process Variations, Including Spatial Correlations," in *Proc. 42nd Design Automation Conf.*, June 2005, pp. 523–526.

[23] R. Y. Chen, N. Vijaykrishnan, and M. J. Irwin, "Clock Power Issues in System-on-a-Chip Designs," in *Proceedings IEEE Computer Society Workshop*, 1999, pp. 48–53.

[24] Z. Chen, M. Johnson, L. Wei, and K. Roy, "Estimation of Standby Leakage Power in CMOS circuits Considering Accurate Modeling of Transistor Stacks," in *Proceedings of the 1998 International symposium on Low Power Electronics and Design*, Aug 1998, pp. 239–244.

[25] Z. Chen, M. Johnson, L. Wei, and W. Roy, "Estimation of Standby Leakage Power in CMOS Circuit Considering Accurate Modeling of Transistor Stacks," *Proc. International Symp. on Low Power Electronics and Design*, pp. 239–244, Aug. 1998.

[26] Z. Chen, K. Roy, and T.-L. Chou, "Efficient Statistical Approach to Estimate Power Considering Uncertain Properties of Primary Inputs," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 3, pp. 484–492, Sep 1998.

[27] M. A. Cirit, "Estimating Dynamic Power Consumption of CMOS Circuits," in *Proceedings of IEEE International Conference on Computer-Aided Design*, Nov. 1987, p. 534537.

[28] A. Davoodi and A. Srivastava, "Probabilistic Dual-Vth Leakage Optimization Under Variability," in *Proc. International Symp. Low Power Electronics and Design*, 2005, pp. 143–168.

[29] C.-S. Ding, C.-Y. Tsui, and M. Pedram, "Gate-Level Power Estimation Using Tagged Probabilistic Simulation ," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 11, pp. 1099–1107, Nov. 1998.

[30] M. S. Elrabaa, I. S. Abu-Khater, and M. I. Elmasry, *Advance Low-Power Digital Circuit Techniques*. Boston: Springer, 1997.

[31] J. Frenkil, "A Multi-Level Approach to Low-Power IC Design," *IEEE Spectrum*, vol. 35, no. 2, pp. 54–60, 1998.

[32] S. Garg, S. Tata, and R. Arunachalam, "Static Transition Probability Analysis Under Uncertainty," in *Proc. IEEE International Conference on Computer Design*, 2004, pp. 380–386.

[33] B. J. George, D. Gossain, S. C. Tyler, M. G. Wloka, and G. K. Yeap, "Power Analysis and Characterization for Semi-Custom Design," in *Proc. Int. Workshop on Low Power Design*, Apr. 1994, pp. 215–218.

[34] H. Grimes, "Reconvergent Fanout Analysis of Bounded Gate Delay Faults," Master's thesis, Auburn University, Aug. 2008. Dept. of ECE.

[35] H. Grimes and V. D. Agrawal, "Analyzing Reconvergent Fanouts in Gate Delay Fault Simulation," in *Proc. 17th IEEE North Atlantic Test Workshop*, May 2008, pp. 98–103.

[36] R. X. Gu and M. I. Elmasry, "Power Dissipation Analysis and Optimization of Deep Submicron Circuits," *IEEE Journal of Solid-State Circuits*, pp. 707–713, May 1996.

[37] S. Hassoun, "Critical Path Analysis Using a Dynamically Bounded Delay Model," in *Proc. 37th Design Automation Conf.*, 2000, pp. 260–265.

[38] J. Hayes, "An Introduction to Switch-Level Modeling," *IEEE Design and Test of Computers*, vol. 4, no. 4, pp. 18–25, 1987.

[39] N. Hedenstierna and K. O. Jeppson, "CMOS Circuit Speed and Buffer Optimization," *IEEE Transactions on CAD*, vol. 6, no. 2, pp. 270–281, Mar. 1987.

[40] F. Hu, *Process-Variation-Resistant Dynamic Power Optimization for VLSI Circuits*. PhD thesis, Auburn University, May 2006. Dept. of ECE.

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

[42] F. Hu and V. D. Agrawal, "Enhanced Dual-Transition Probabilistic Power Estimation With Selective Supergate Analysis," *Proc. IEEE International Conference on Computer Design*, pp. 366–369, Oct. 2005.

[43] F. Hu and V. D. Agrawal, "Input-Specific Dynamic Power Optimization for VLSI Circuits," in *Proc. Int. Symp. on Low Power Electronics and Design (ISLPED'06)*, Oct. 2006, pp. 232–237.

[44] C. X. Huang, B. Zhang, A.-C. Deng, and B. Swirski, "The Design and Implementation of PowerMill," in *Proc. Int. Workshop Low Power Design*, Apr. 1995, pp. 105 –110.

[45] R. C. Jaeger and T. N. Blalock, *Microelectronic Circuit Design*. McGraw Hill, second edition, 1997.

[46] S. M. Kang, "Accurate Simulation of Power Dissipation in VLSI Circuits," *IEEE Journal of Solid-State Circuits*, vol. 21, no. 5, pp. 889–891, Oct 1986.

[47] S. P. Khatri, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, *Cross - Talk Noise Immune VLSI Design Using Regular Layout Fabrics*. Boston: Springer, 2001.

[48] J.-T. Kong, S. Z. Hussain, and D. Overhauser, "Performance Estimation of Complex MOS Gates," *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol. 44, no. 9, pp. 785–795, Sep 1997.

[49] T. H. Krodel, "Power Play-Fast Dynamic Power Estimation Based on Logic Simulation ," *Proc. IEEE International Conference on Computer Design*, pp. 96–100, Oct 1991.

[50] J. C. Ku, M. Ghoneima, and Y. Ismail, "The Importance of Including Thermal Effects in Estimating the Effectiveness of Power Reduction Techniques," in *Proc. IEEE Custom Integrated Circuits Conference*, Sept 2005, pp. 301–304.

[51] R. Kumar and C. P. Ravikumar, "Leakage Power Estimation for Deep Submicron Circuits in an ASIC Design Environment," in *Proc. IEEE Alessandro Volta Memorial Workshop*, 2002, pp. 45–50.

[52] K. N. Lalgudi, D. Bhattacharya, and P. Agrawal, "Architecture of a Min-Max Simulator on MARS," in *Proc. International Conf. VLSI Design*, Jan. 1993, pp. 246–249.

[53] W. K. C. Lam and R. K. Bryton, *Timed Boolean Functions: A Unified Formalism for Exact Timing*. Springer, 1994.

[54] M. Linderman and M. Leeser, "Simulation of Digital Circuits in the Presence of Uncertainty," in *Proc. of International Conf. Computer-Aided Design*, 1994, pp. 248–251.

[55] Y. Lu, *Power and Performance Optimization of Static CMOS Circuits with Process Variation*. PhD thesis, Auburn University, Aug. 2007. Dept. of ECE.

[56] Y. Lu and V. D. Agrawal, "Leakage and Dynamic Glitch Power Minimization Using Integer Linear Programming for V_{th} Assignment and Path Balancing," in *Proc. Power and Timing Modeling, Optimization and Simulation Workshop (PATMOS'05)*, Sept. 2005, pp. 217–226.

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

[58] E. J. McCluskey, "Transients in Combinational Logic Circuits," in R. H. Wilcox and W. C. Mann, editors, *Redundancy Techniques for Computing Systems*, pp. 9–46, Washington, D.C.: Spartan Books, 1962.

[59] P. C. McGeer and R. K. Brayton, *Integrating Functional and Temporal Domains in Logic Design*. Springer, 1991.

[60] G. Merrett and B. M. Al-Hashimi, "Leakage Power Analysis and Comparison of Deep Submicron Logic Gates," in *Proc. 14th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, sep 2004.

[61] 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. International Symp. Low Power Electronics and Design*, Aug. 2003, pp. 172–175.

[62] L. W. Nagel, *SPICE2, A Computer Program to Simulate Semiconductor Circuits*. PhD thesis, University of California, Electronics Research Laboratory, Berkeley, California, May 1975. Dept. of EECS.

[63] F. N. Najm, "Transition Density: A New Measure of Activity in Digital Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 2, pp. 310–323, Feb. 1993.

[64] F. N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," *IEEE Trans. VLSI Systems*, vol. 2, no. 4, pp. 446–455, Dec. 1994.

[65] F. N. Najm, "Power Estimation Techniques for Integrated Circuits," *Proc. IEEE/ACM International Conf. on Computer-Aided Design*, pp. 492–499, Nov 1995.

[66] F. N. Najm, R. Burch, P. Yang, and I. N. Hajj, "CREST - A Current Estimator for CMOS Circuits," in *Proceedings of IEEE International Conference on Computer-Aided Design*, Nov. 1988, p. 204207.

[67] F. N. Najm, R. Burch, P. Yang, and I. N. Hajj, "Probabilistic Simulation for Reliability Analysis of CMOS VLSI Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 4, pp. 439–450, Apr. 1990.

[68] N. Nicolici and B. M. Al-Hashimi, *Power-Constrained Testing of VLSI Circuits*. Springer, 2003.

[69] S. Nikolaidis and A. Chatzigeorgiou, "Analytical Estimation of Propagation Delay and Short-Circuit Power Dissipation in CMOS Gates," *International Journal of Circuit Theory and Applications*, vol. 27, pp. 39–2, 1999.

[70] Y. Park and E. Park, "Statistical Power Estimation of CMOS Logic Circuits with Variable Errors," *Electronics Letters*, vol. 34, no. 11, pp. 1054–1056, May 1998.

[71] C. Piguet, "Circuit and Logic Level Design," in W. Nebel and J. Mermet, editors, *Low Power Design in Deep Submicron Electronics*, pp. 105–133, Springer, 1997.

[72] Q. Qiu, Q. Wu, and M. Pedram, "Maximum power estimation using the limiting distributions of extreme order statistics," in *Proc. Design Automation Conference*, June 1998, pp. 684–689.

[73] R. Radjassamy and J. D. Carothers, "Simulation-based Power Estimation for Low Power Eesigns: A Fractal Approach," *Simulation*, vol. 72, no. 5, pp. 320–326, 1999.

[74] T. Raja, "A Reduced Constraint Set Linear Program for Low-Power Design of Digital Circuits," Master's thesis, Rutgers University, Mar. 2002. Dept. of ECE.

[75] T. Raja, *Minimum Dynamic Power CMOS Design with Variable Input Delay Logic*. PhD thesis, Rutgers University, May 2004. Dept. of ECE.

[76] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Minimum Dynamic Power CMOS Circuit Design by a Reduced Constraint Set Linear Program," in *Proc. 16th International Conf. VLSI Design*, Jan. 2003, pp. 527–532.

[77] T. Raja, V. D. Agrawal, and M. L. Bushnell, "CMOS Circuit Design for Minimum Dynamic Power and Highest Speed," in *Proc. 17th International Conf. VLSI Design*, Jan. 2004, pp. 1035–1040.

[78] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Variable Input Delay CMOS Logic Design for Low Dynamic Power Circuits," in *Proc. Power and Timing Modeling, Optimization and Simulation Workshop (PATMOS'05)*, Sept. 2005, pp. 436–445.

[79] R. M. Rao, J. L. Burns, A. Devgan, and R. B. Brown, "Efficient Techniques for Gate Leakage Estimation," *Proc. International Symp. on Low Power Electronics and Design*, pp. 100–103, Aug. 2003.

[80] S. Roy, P. P. Chakrabarti, and P. Dasgupta, "Bounded Delay Timing Analysis Using Boolean Satisfiability," in *Proc. International Conf. VLSI Design*, Jan. 2007, pp. 295–302.

[81] A. Salz and M. A. Horowitz, "IRSIM: An Incremental MOS Switch-Level Simulator," in *Proc. 26th Design Automation Conf.*, June 1989, p. 173178.

[82] C. J. Seger, "A Bounded Delay Race Model," in *Proc. of the IEEE International Conf. Computer Aided Design*, Nov. 1989, pp. 130–133.

[83] S. C. Seth and V. D. Agrawal, "A New Model for Computation of Probabilistic Testability in Combinational Circuits," *INTEGRATION, The VLSI Journal*, vol. 7, pp. 49–75, 1989.

[84] J. Sheu, "BSIM: Berkeley Short-Channel IGFET Model for MOS transistors," *IEEE Journal of Solid-State Circuits*, vol. 22, no. 4, pp. 558–566, Aug 1987.

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

[86] A. Srivastava, D. Sylvester, and D. Blaauw, "Statistical Optimization of Leakage Power Considering Process Variations Using Dual-Vth and Sizing," in *Proc. 41st Design Automation Conf.*, June 2004, pp. 783–787.

[87] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical Analysis and Optimization for VLSI: Timing and Power*. Boston: Springer, 2005.

[88] Synopsys, *HSPICE User's Manual*, w 2005.03 edition, 2005.

[89] R. Tjarnstrom, "Power Dissipation Estimate by Switch Level Simulation of CMOS Circuits," *Proc. IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 881–884, May 1989.

[90] C.-Y. Tsui, M. Pedram, and A. Despain, "Efficient Estimation of Dynamic Power Consumption Under a Real Delay Model," *Proc. IEEE/ACM International Conference on Computer-Aided Design*, pp. 224–228, Nov. 1993.

[91] S. Uppalapati, "Low Power Design of Standard Cell Digital VLSI Circuits," Master's thesis, Rutgers University, Mar. 2004. Dept. of ECE.

[92] S. Uppalapati, M. L. Bushnell, and V. D. Agrawal, "Glitch-Free Design of Low Power ASICs Using Customized Resistive Feedthrough Cells," in *Proc. 9th VLSI Design & Test Symp. (VDAT'05)*, Aug. 2005, pp. 41–49.

[93] H. J. M. Veendrick, "Short-Circuit Dissipation of Static CMOS Circuitry and its Impact on the Design of Buffer Circuits," *IEEE Journal of Solid-State Circuits*, vol. 19, no. 4, pp. 468–473, Aug. 1984.

[94] S. R. Vemuru and N. Scheinberg, "Short-Circuit Power Dissipation Estimation for CMOS Logic Gates," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 41, no. 11, pp. 762–765, Nov. 1994.

[95] C.-Y. Wang, T.-L. Chou, and K. Roy, "Maximum Power Estimation for CMOS Circuits Under Arbitrary Delay Model," in *Proc. IEEE International Symp. Circuits and Systems*, May 1996, pp. 763–766.

[96] Q. Wang and S. B. K. Vrudhula, "On Short Circuit Power Estimation of CMOS Inverters," *Proc. IEEE International Conference on Computer Design*, pp. 70–75, Oct. 1998.

[97] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and System Perspective*. Addison Wesley, third edition, 2004.

[98] Q. Wu, Q. Qiu, and M. Pedram, "Estimation of Peak Power Dissipation in VLSI Circuits Using the Limiting Distributions of Extreme Order Statistics," *IEEE transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 20, no. 8, p. 942.

[99] M. G. Xakellis and F. N. Najm, "Statistical Estimation of the Switching Activity in Digital Circuitry," *Proc. 31st Design Automation Conf.*, pp. 728–733, June 1994.

[100] G. Y. Yacoub and W. H. Ku, "An Accurate Simulation Technique for Short Circuit Power Dissipation Based on Current Component Isolation," in *Proc. of the International Symposium on Circuits and Systems*, 1989, pp. 1157–1161.

[101] G. K. Yeap, *Practical Low Power Digital VLSI Design*. Springer, 1998.