ULTRA LOW POWER ROBUST DESIGN FOR NANOMETER CMOS TECHNOLOGY: PROCESS, CIRCUIT AND ARCHITECTURE PERSPECTIVES

by

Ruth Ann Wang

Memorandum No. UCB/ERL M05/3

20 December 2004

ULTRA LOW POWER ROBUST DESIGN FOR NANOMETER CMOS TECHNOLOGY: PROCESS, CIRCUIT AND ARCHITECTURE PERSPECTIVES

by

Ruth Ann Wang

Memorandum No. UCB/ERL M05/3

20 December 2004

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Ultra Low Power Robust Design for Nanometer CMOS Technology

Process, Circuit and Architecture Perspectives

Ruth Ann Wang
University of California, Berkeley
Department of Electrical Engineering and Computer Sciences

December 20, 2004

Ultra Low Power Robust Design for Nanometer CMOS Technology: Process, Circuit and Architecture Perspectives

by Ruth Ann Wang

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II.**

Approval for the Report and Comprehensive Examination:

Committee:

Professor Jan Rabaey
Research Advisor

12/13/04
Date

* * * * * *

R. K. Brayton
Professor Robert Brayton
Second Reader

12/15/04
Date

Ultra Low Power Robust Design for Nanometer CMOS Technology:
Process, Circuit and Architecture Perspectives

by
Ruth Ann Wang

Abstract

VLSI designs for wireless applications have increasingly relied on aggressive voltage and device size scaling in order to achieve reductions in area, cost and power dissipation. However, as the power supply voltage decreases and device sizes scale into the nanometer regime, fluctuations in environmental and physical factors become more difficult to control. Variations in supply voltage, transistor gate length and threshold voltage increase in proportion to their respective nominal values, causing a widened overall distribution of values for all performance metrics, particularly gate propagation delay. Consequently, traditional worst case design leads to prohibitively large delay overheads at ultra low supply voltages. This work investigates a novel timing methodology that designs for variation-induced timing errors, using robust design techniques to ensure proper system functionality. Monte Carlo simulation environments are used to simulate variability in circuit performance metrics by subjecting process and operating parameters to controlled fluctuation levels. The resulting robustness of circuits is evaluated and techniques of supply and threshold voltage scaling are studied to explore trade-offs between yield and energy. Furthermore, individual parameter contributions to delay variability are isolated in order to identify potential sources of improvement in manufacturing processes. Finally, a fault tolerant approach to finite state machine design is proposed and studied using MVSIS, in which transistor-level timing errors are modeled as faulty system behavior.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor, Professor Jan Rabaey, for his guidance and support throughout the years.

I am grateful to Professor Robert Brayton and Alan Mishchenko for offering EE290N during the Spring of 2004 and spending office hours twice a week teaching me everything I know about MVSIS.

Thanks to members of the YODA group at the Berkeley Wireless Research Center: Yu Cao, Huifang Qin, Liang-Teck Pang, Paul Friedberg and Professor Andrei Vladimirescu, for their ideas and invaluable feedback.

To my mentors Kerry Bernstein and Dale Pearson.

To Nancy B. Green, who still has the binary half-adder that I built in my high school physics class.

Thank you Mom, Dad and Chris, for a lifetime of love and support.

and Brian Otis, for being a source of constant inspiration, always.

Chapter 1

Motivation: Ultra Low Power

Despite forecasts in the 1970s proclaiming that the scaling of integrated circuits (ICs) would not succeed beyond critical dimensions of 0.5μm [1], the state of the art has accelerated well into the nanometer regime with unprecedented momentum. Present-day low power application drivers, such as truly ambient intelligent systems and highly energy-efficient sensor networks, have increasingly pushed technology innovation and motivated research thrusts to realize novel design techniques. In particular, the VLSI designs for these applications have combined aggressive device and voltage scaling techniques to achieve reductions in power, area and cost, with extremely high levels of integration.

The challenge of reducing the power consumption of a system is a primary concern for designers, and is especially critical as device sizes and form factors continue to scale. The total power dissipation in a digital CMOS circuit design is attributed to two primary sources of current flow: static leakage current and active switching current. Leakage current is lost through resistive paths between voltage supply and ground, leading to static power dissipation, which may be described as follows [2, 3]:

$$P_{static} \propto I_s \cdot exp\left[\frac{-(V_{th} - \gamma V_{dd})}{S}\right] \cdot V_{dd} \tag{1.1}$$

where I_s is the zero-threshold leakage current, V_{th} is the threshold voltage, V_{dd} is the supply voltage, S is the subthreshold slope and γ is a fitted parameter modeling the effects of drain-induced barrier lowering (DIBL).

While the circuit is active and signals are dynamically switching, current is alternately drawn from and pulled into the supply rails in order to charge and discharge capacitive loads. The total active power dissipated is the amount of energy consumed for each switching operation, with switching energy defined as follows [3]:

$$E_{active} = \alpha \cdot C_L \cdot V_{dd}^2 \tag{1.2}$$

where α represents the average activity factor of gates that compose the design, C_L represents the total load capacitance, and V_{dd} is the operating supply voltage. The total dynamic power consumed is determined by the frequency f with which the switching operations are performed:

$$P_{dynamic} = \alpha \cdot C_L \cdot V_{dd}^2 \cdot f \tag{1.3}$$

Among the diverse field of applications for ultra low power CMOS designs, one primary application of this work is in the emerging space of wireless sensor networks. These low power systems may be used for a wide range of military, medical and environmental monitoring applications, and are the focus of study for researchers at the Berkeley Wireless Research Center at the University of California, Berkeley. The behavioral model of an ultra low power sensor node contains two states: idle and processing. Each node is primarily in the idle state until an event is detected, such as the arrival of a data packet, at which point the circuits are activated for data processing. After this burst of activity, the system returns again to its idle state and

remains there until the next event occurs. Because wireless sensor networks are characterized by long periods of inactivity, highly efficient power management techniques may be implemented that eliminate static leakage current (e.g. disconnecting circuits from the power supply [4]) while the system idles. Therefore, while total power consumption is the sum of both static and dynamic components, this work focuses on reducing dynamic power dissipation because it is assumed that high levels of static leakage power are mitigated by system-level techniques.

It is clear from Equations (1.1) and (1.3) that one of the most effective techniques for reducing total power in a CMOS design is by reducing the supply voltage. Specifically, when considering only the dynamic component, power dissipation falls quadratically with V_{dd}, suggesting that significant reductions in power consumption may be achieved with relatively small decreases in the supply voltage.

The technique of scaling supply voltage in low power designs is combined with the scaling of device sizes, which reduces capacitive loading as well as circuit area, further reducing power consumption and form factors. Near- and long-term trends in the scaling of supply voltage and effective gate length (L_{eff}) for digital circuits in low power operation have been projected by the International Technology Roadmap for Semiconductors (ITRS) [5] and are shown in Figure 1.1.

These scaling predictions indicate that within the next two decades, supply voltages will reduce to 500mV, while effective gate lengths will shrink as low as 9nm. This combination of aggressive scaling techniques, with critical dimensions nearing the atomic scale of angstroms (10^{-10} m), induces increased variation in circuit designs, a problem that remains largely unresolved. Topping the 2003 ITRS list of most difficult challenges for sub-20nm CMOS transistor designs are the fundamental issues associated with atomic-level, statistical process fluctuations; process imperfections become more difficult to control as physical parameters scale, leading to a wider

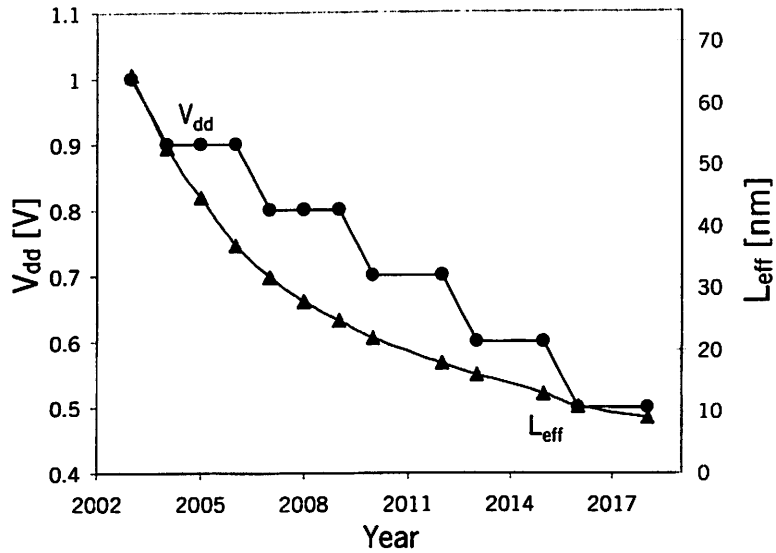


Figure 1.1: ITRS scaling projections for low power operation (2003).

spread of manufactured device parameters. Further contributing to this physical variability source are environmental factors, such as increased noise in power supply voltages and changes in operating temperature. According to industrial predictions and observations in deep submicron designs, variations in L_{eff} were projected to increase from 30% to nearly 50% across the span of three technology generations [6]. Variations in V_{dd} and V_{th} were also projected to rise, both from 10% to 15%. The combination of increased variability in both physical device parameters and circuit operational conditions leads to a widening distribution of values for all performance metrics.

With circuit behavior increasingly less predictable as technology scales, a new, robust circuit design methodology is of paramount importance to the success of future nanometer designs. A *robust* product or process is defined as:

> [one that] performs properly even in the presence of uncontrolled variation

that may affect performance, such as manufacturing variations, operating conditions, and product deterioration. [7]

When applied to ultra low power circuits, robust design refers to the variation-aware design methodologies required to ensure proper functionality across all worst-case parameter corners. To ensure the most effective approach to robust design, these techniques should be considered at all layers of the system design hierarchy: from low levels of manufacturing and process control, to intermediate levels of transistor, circuit and logic design, ultimately reaching the highest levels of architecture, algorithm, and system organization.

The objective of this work is to investigate the robust design of ultra low power CMOS circuits, which operate under aggressively scaled supply voltages and comprise nanometer-scale transistors. The research is performed while considering process, circuit and architecture perspectives, and is described in three parts. First, the impact of parameter variations on circuit performance is investigated in Chapter 2 using a SPICE simulation environment in 130nm bulk CMOS technology. Next, Chapter 3 details the extension of this study to an industrial, 90nm partially depleted silicon-on-insulator (pd-SOI) technology, in which robust design is also approached from a manufacturing and process control perspective. Chapter 4 explores the impacts of parameter variability on higher layers of the design hierarchy, with a focus on techniques for robust finite state machine design. Finally, Chapter 5 offers concluding remarks and directions for future work.

Chapter 2

Process/Circuits Co-Design: Energy-Delay Tradeoffs

An understanding of the nature of errors that may occur in aggressively scaled, ultra low power digital circuits is crucial for building the foundations of a robustness study. Thus, the first step is to simulate a realistic environment in which circuits are operated under reduced supply voltages and subjected to variations in physical parameter values, in order to induce errors from fundamental sources. This study is performed in a standard bulk CMOS technology, and compares the robustness of a number of representative circuit blocks of varying complexity and implemented in a variety of logic styles. Physical and environmental sources of parameter variation are now introduced.

2.1 Sources of Delay Variability

The propagation delay of a transistor is related to its operating supply and threshold voltages, as well as the physical process parameters that define its intrinsic logic evaluation capability. Therefore, the scaling of operating voltage and physical dimensions affects raw values of transistor performance to a first-order, and also causes higher order effects, which manifest as performance variation. The extent to which delay is

sensitive to these variations is extremely difficult to predict accurately; this section provides a basic intuition for understanding these relationships.

2.1.1 Delay Sensitivity to Operating Voltage

While the reduction of supply voltage leads to quadratic savings in active energy and power dissipation, it is known to increase both delay and delay variability [8]. To investigate this phenomenon, the delay of a logic gate is modeled by the following equation, which is based on the alpha power law [2, 3]:

$$\tau_d \propto \frac{V_{dd}}{(V_{dd} - V_{th})^{\alpha}} \tag{2.1}$$

where V_{dd} is the supply voltage, V_{th} is the operating threshold voltage and α is a fitted parameter with a value between one and two, modeling the effects of velocity saturation.

It is clear from this relationship that raw values of delay will increase with reduced supply voltages, because the order of the denominator term is greater than the order of the numerator. Moreover, while the increased propagation delay poses challenges for maintaining competitive clock frequencies in future designs, the ability to control the range in which the delay varies is a substantially more crucial challenge. Tolerating an absolute, fixed delay offset is a trivial task when compared with designing for a spread of delays that may vary as widely as the nominal delay itself. To gain insight into the extent to which reduced supply voltage affects delay variability, the following definition for the sensitivity of gate delay with respect to V_{th} is presented:

$$S_{\tau_d}^{V_{th}} \triangleq \frac{\partial \tau_d}{\partial V_{th}} \tag{2.2}$$

Solving for the partial derivative of the gate delay with respect to V_{th} yields the following sensitivity of delay to threshold voltage:

$$\begin{aligned} \frac{\partial \tau_d}{\partial V_{th}} &\propto -\alpha V_{dd}(V_{dd} - V_{th})^{-\alpha-1}(-1) \\ &= \frac{\alpha V_{dd}}{(V_{dd} - V_{th})^{\alpha+1}} \end{aligned} \tag{2.3}$$

$$\therefore \quad S_{\tau_d}^{V_{th}} \propto \frac{\alpha V_{dd}}{(V_{dd} - V_{th})^{\alpha+1}} \tag{2.4}$$

As V_{dd} is lowered, the denominator term of this sensitivity relationship decreases at a greater rate than the numerator term, leading to exponentially higher V_{th} sensitivity at low supply voltages. This result confirms a known challenge to the continued success of digital design for future scaled generations: not only do absolute delay values increase with lowered supply voltages, but so does the variability of those delays. Adding to these variability levels are variations in physical parameters, which are now discussed.

2.1.2 Delay Sensitivity to Physical Process Parameters

The threshold voltage of a transistor is determined by physical parameters set by the manufacturing process and is affected by imperfections in process steps. The following expression is used to estimate the standard deviation of the manufactured threshold voltage from its mean design value [9]:

$$\sigma_{V_{th}} \propto \frac{t_{ox}\sqrt[4]{NT}}{\sqrt{W_{eff}L_{eff}}} \tag{2.5}$$

where t_{ox} is the thickness of the gate oxide, N is the channel doping density, T is the absolute temperature, and W_{eff} and L_{eff} are the effective width and length of the

transistor, respectively. Variations in these manufactured physical parameters induce variability in V_{th}, further contributing to variability in gate delay, as seen in Equation (2.4). Based upon the above delay sensitivity analysis, parameter variations in a design are attributed to two sources: fluctuations in environmental conditions (V_{dd}, T) and imperfections in physical device structure (L_{eff}, t_{ox}, W and V_{th}). Interactions between these distinct variation sources produce an increased spread of delays, relative to their nominal values. Methods for reducing the extent of this variability, including a metric for quantifying the impact of delay variability on performance-based yield, are the basis of this robustness study.

2.2 Previous Research

Previous work in the field of robust circuit design serves as background knowledge and provides directions for further study. This related body of research includes comparisons of performance between logic evaluation styles, studies on circuit delay variability for a range of process variations, and techniques for achieving robust low power design using threshold voltage optimization. After related work is introduced, the contributions of this work and corresponding experimental setup are discussed.

While numerous studies have compared circuits across complexity and logic evaluation style for metrics such as performance, power, and area, few have included a discussion of inherent robustness of logic topology to process parameter variations. In [10], standard delay-power tradeoffs were studied for various circuits, including full adders and 2-input NAND gates, implemented in both static CMOS and pass-gate logic. Results from HSPICE simulations showed static CMOS to be the more favorable topology for use in low power design, due to significant gains in power dissipation that outweigh its comparably lower performance. Although this work il-

lustrated a rigorous technique for evaluating the tradeoff between power and speed in low power design, it did not include parameter variability as a significant factor affecting circuit performance. Thus, one direction for further investigation is to compare the relative robustness of various logic evaluation styles, when these designs are subjected to variations in operating and physical parameters.

Recent efforts to study the impacts of increased device parameter variations on circuit performance have treated only relatively simple circuits, typically implemented only in static CMOS. In [11], a Monte Carlo analysis was conducted for a 2-stage inverter chain in order to study the implications of worst-case variation for several physical and environmental parameters (including L_{eff}, t_{ox}, V_{dd} and V_{th}). The resulting analysis confirmed the underlying challenge for scaled designs in the nanometer regime: as technology parameters are scaled, their variations increase relative to nominal values, thus exacerbating delay variability. While techniques of aggressive buffer insertion and careful wire sizing were suggested as a means for controlling excessive variability, circuit-level timing consequences were not discussed. An extension of this work was conducted in [12], with the inclusion of the more complex NAND chain in a similar variability investigation. However, all circuits in this study were implemented in static CMOS and thus the impact of circuit topology on delay distribution was not considered. Furthermore, while both studies confirmed the trend of increasing global delay variations with device size scaling, neither quantified the extent to which the increased variability may affect circuit timing methodologies.

Guidelines for achieving minimum power dissipation in a circuit while maintaining robustness to parameter variation were set in [13]. The technique of scaling V_{th} along with V_{dd} was found to improve performance under low voltage conditions. Furthermore, longer effective channel lengths were chosen in order to reduce variations in V_{th} and thus lower delay variability. The optimal voltage ranges used in this work

were relatively high; V_{dd} was scaled to a minimum of 600mV while V_{th} values were chosen between 340mV – 450mV. Because these threshold and supply voltages were maintained near their nominal values, the spread of delay values was sufficiently contained such that a worst-case timing methodology was reasonable for determining the clock frequency. Thus, aggressively scaled voltages and their impact on increased delay variability were not explored.

Given the unknown design space and questions unaddressed by existing research, the focus of this work is to explore the field of robust circuit design for dramatically scaled voltages, across circuits of varying complexity and logic topology. Therefore, a set of representative circuits is designed and subjected to exhaustive Monte Carlo simulations, and the effects of parameter variations are investigated. The remainder of this chapter is organized as follows. Section 2.3 describes the simulation setup, including technology specifications and circuits under study. Section 2.4 discusses the statistical model used to analyze the simulation results and presents a performance-based yield metric to quantify tradeoffs between energy and delay for a given circuit. Results are presented in Section 2.5 and Section 2.6 concludes the analysis.

2.3 Experimental Setup

The experimental setup for this work is described in two parts: the Monte Carlo simulation framework and the various circuits under study.

2.3.1 Monte Carlo Simulation Framework

A Monte Carlo simulation is a method for simulating a model for a process whose behavior cannot be or is not easily determined from a closed-form expression. The values of parameters affecting the process are uncertain and vary according to a

known distribution, such that the output is not a fixed value that may be predicted with high accuracy, but rather one among a statistical spread of possible values. An analysis of the space of all outcomes is conducted by running an exhaustive number of simulations; for each simulation, all parameter values are drawn randomly from their respective distributions.

The resulting collection of output data provides statistical insights into the nominal expected output value (mean) and how well that nominal value may be predicted (variance). The accuracy of the simulation is increased by sampling parameter values at minimally sized intervals, while iterating through all parameter combinations.

As discussed in Section 2.1, the spread of performance abilities for a given circuit is influenced by imperfections in the manufacturing process, as well as fluctuations in environmental conditions. The ranges of these variations are input to a Monte Carlo simulation, which is applied in this work to characterize the performance distribution of a number of representative circuits.

Technology Specifications

Simulations in this work are run in a 130nm bulk CMOS technology, using the industry standard BSIM3v3 device model [14], with nominal values and variation ranges set by the Berkeley Predictive Technology Model (BPTM) [15]. The supply voltage V_{dd} and threshold voltage V_{th} are discretized into a range of incremental design values: V_{dd} is scaled from 1.2V to 300mV in steps of 100mV, while V_{th} is decremented from 240mV to 40mV in 50mV steps.

Variations in the threshold voltage of a transistor are attributed to two primary sources: random fluctuations in atom concentrations during channel doping, and variations in channel lengths, which induce large threshold variations for short channels. The sharp roll-off in V_{th} with decreasing values of L_{eff} is due to DIBL and the short

channel effect (SCE); the combined effects of these two physical parameters influence delay variability significantly more than contributions from W, t_{ox} and T [12]. Small changes in W do not cause comparably significant changes in delay, and t_{ox} has historically been one of the most well controlled manufacturing parameters because of its critical impact on transistor performance. Furthermore, while variations in operating temperature certainly affect nominal performance metrics, they do not significantly alter the shape of the performance distribution. Therefore, V_{dd}, V_{th} and L_{eff} are chosen as the variable process parameters in this study, while the remainder are fixed at their nominal values.

In order to decouple the variation sources affecting threshold voltage, and thus reconcile the interdependencies between V_{th} and other variable parameters, the SPICE simulation environment calculates the operating threshold voltage as the sum of distinct component contributions. Equation (2.6) describes the four primary components, as dictated by the BSIM model [14]:

$$V_{th,operating} \simeq V_{th,intrinsic} + \Delta V_{th,HALO} - \Delta V_{th,DIBL} + \Delta V_{th,BIAS} \tag{2.6}$$

where

- $V_{th,intrinsic} = f(N_{channel}, \phi_S, \phi_M, t_{ox})$
- $\Delta V_{th,HALO} = f(N_{halo}, L_{eff})$
- $\Delta V_{th,DIBL} = f(V_{ds}, L_{eff})$
- $\Delta V_{th,BIAS} = f(V_{bs})$

It is clear from the above relations that the intrinsic value of V_{th}, also known as the long channel value, depends only on the channel doping concentration, the work functions ϕ_S and ϕ_M of silicon and the gate material, and the thickness of the oxide; it is independent of the channel length. The two factors that are dependent upon

Table 2.1: Technology specifications for parameters varied in Monte Carlo simulation.

Parameter	Mean	3σ/mean
$L_{eff,\ nmos}$	71 nm	15%
$L_{eff,\ pmos}$	80 nm	15%
$V_{th,\ nmos}$	240 mV	15%
$V_{th,\ pmos}$	-340 mV	15%
V_{dd}	1.2 V	10%

L_{eff} are $\Delta V_{th,HALO}$, which describes the rising slope of V_{th} due to HALO doping, and $\Delta V_{th,DIBL}$, which captures the effect of V_{th} lowering as a function of channel length and voltage between the drain and source. The final component, $\Delta V_{th,BIAS}$, describes the "body effect," in which the operating value is further shifted due to a potential difference between the body and source. This term is purely due to biasing and is independent of both channel doping and channel length.

For each SPICE simulation, the *intrinsic* threshold voltage V_{th} and effective channel length L_{eff} are chosen randomly and independently of each other. The appropriate value for the *operating* threshold value is then calculated based upon bias conditions and the corresponding L_{eff} value for that simulation, which affects $\Delta V_{th,DIBL}$ and $\Delta V_{th,BIAS}$. This simulation setup thus individually accounts for the contributions of L_{eff} and $N_{channel}$ to V_{th}, which ensures that the total threshold voltage variation is properly estimated.

Throughout the simulations in this work, V_{th} represents the value of the intrinsic threshold voltage, rather than the operating value. Detailed technology specifications, extracted from the BPTM for all variation sources, are summarized in Table 2.1.

A block diagram representation of the Monte Carlo framework is shown in Figure 2.1. For each of fifty combinations of nominal V_{dd} and V_{th} values, 1000 SPICE simulations are performed on each circuit under study. For each simulation, the exact values of V_{dd}, V_{th} and L_{eff} assigned to the circuit are sampled randomly from each

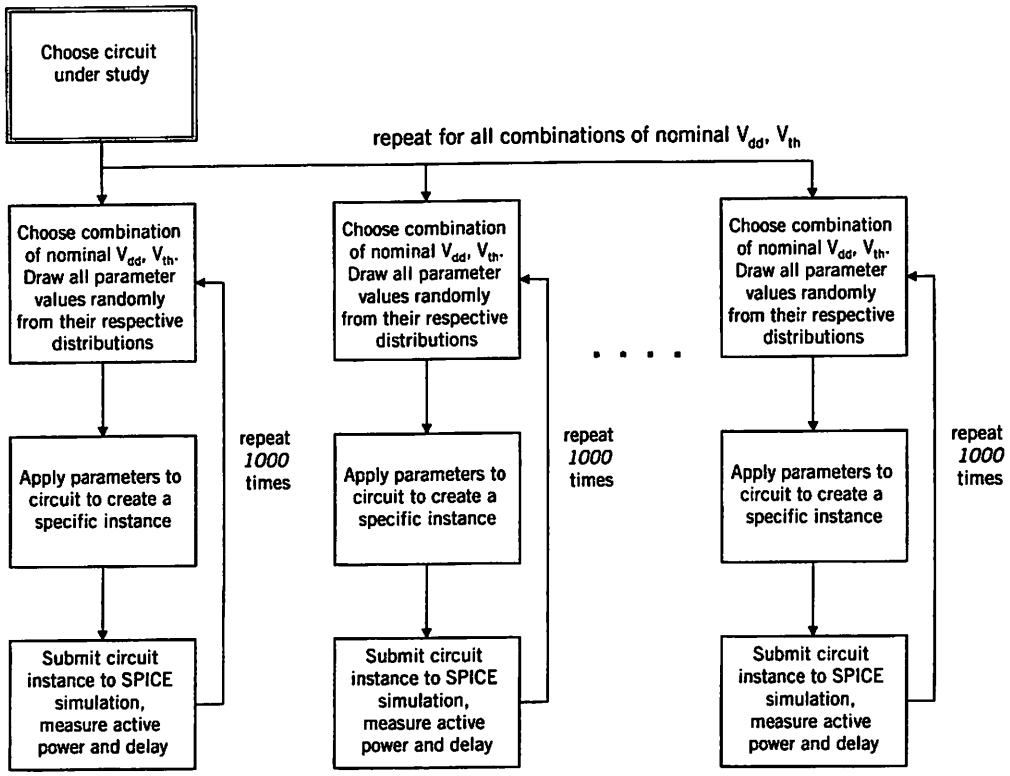


Figure 2.1: Monte Carlo simulation framework for yield-energy study.

corresponding distribution. After these parameter values are assigned and the circuit simulated, the critical path delay and active energy dissipation are measured.

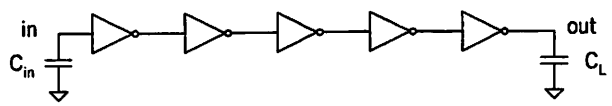
The choice of 1000 simulations per circuit, with ten design values for V_{dd} and five for V_{th}, is somewhat arbitrary but is limited by the total required computation time. Within each of 50,000 circuit instances, one SPICE submission is required per measurement (one each for energy and delay), resulting in a total of 100,000 SPICE simulations. Each simulation requires from 1 – 10 seconds to complete, depending upon the desired level of measurement accuracy, bringing the total computation time to between 28 and 280 hours (12 days) for each circuit. Due to limitations in time and available computing resources, it is undesirable to increase either the number of instances per circuit or the number of design values for V_{dd} and V_{th}.

One significant assumption in this study is that of perfect parameter correlation. Each variable parameter value is drawn from its corresponding distribution and applied to every transistor identically within a single Monte Carlo simulation, implying a correlation coefficient $\rho = 1$. For gate length and supply voltage variations, this assumption is likely valid because the critical logic depths of circuits considered in this study are relatively short (five stages), and variations in L_{eff} and V_{dd} are not likely to differ considerably over such small distances. However, this assumption is somewhat pessimistic when modeling V_{th} variations. In reality, the correlation of intrinsic V_{th} values between adjacent transistors is most likely weaker ($\rho_{V_{th}} < 1$), due to fluctuations in doping levels that are independent of physical proximity. Because statistically uncorrelated distributions combine to produce normal (gaussian) output distributions, the actual overall V_{th} contribution to total variability is likely averaged to a lower value than that estimated by this simulation setup. Certainly as more advanced technology generations are considered, and the dopant fluctuation component of V_{th} variability increases, the assumption of perfect parameter correlation becomes less valid.

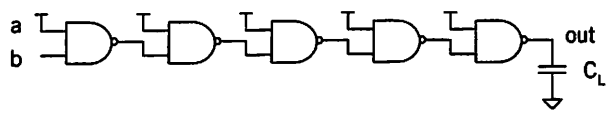
2.3.2 Circuits Under Study

Figure 2.2 shows the types of standard circuits used in this study: a 5-stage inverter chain, a 5-stage 2-input NAND chain and a 4-bit ripple carry adder.

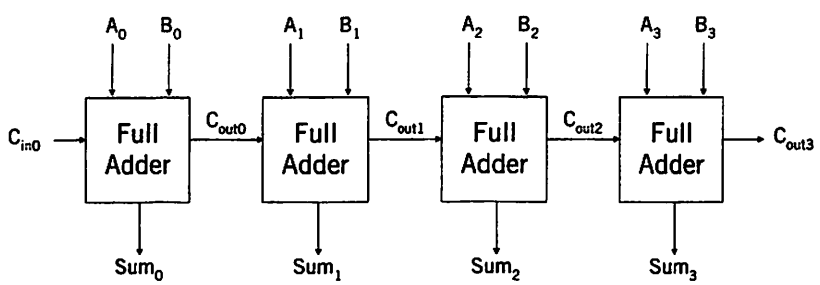
The inverter chain is loaded with a large 1pF capacitor, simulating a buffer with a large signal fanout. The static inverters composing the buffer are progressively sized for optimal delay, using known sizing guidelines for driving long uniform lines [16]. Figures 2.3 and 2.4 illustrate the transistor-level implementations of the NAND and adder circuits, both of which are loaded with relatively smaller capacitive loads of C_L



(a) Five stage inverter chain

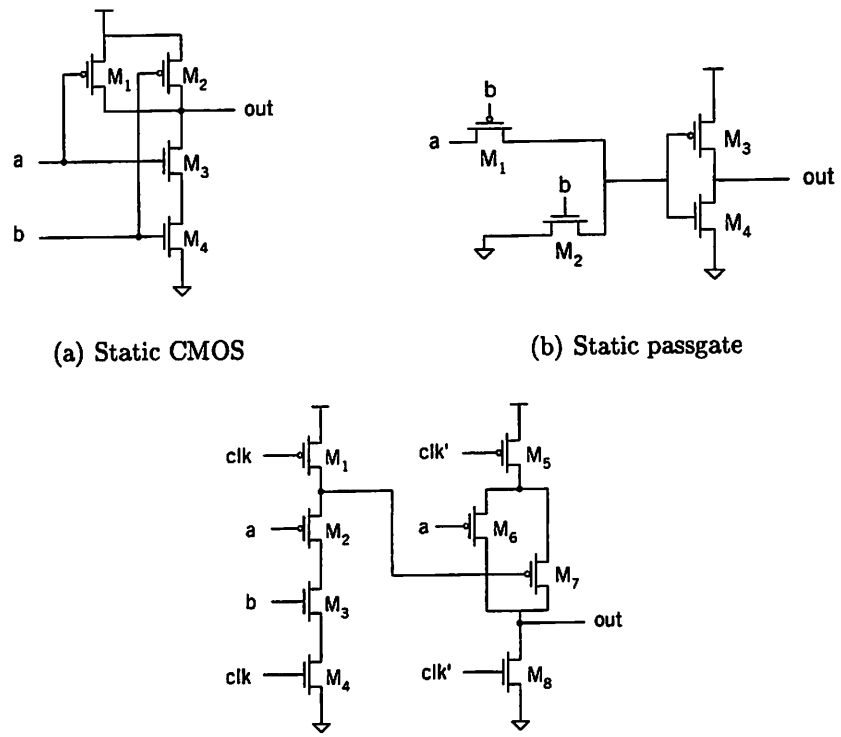


(b) Five stage NAND chain



(c) Four-bit ripple carry adder

Figure 2.2: Block diagrams for representative circuits under study.



(a) Static CMOS

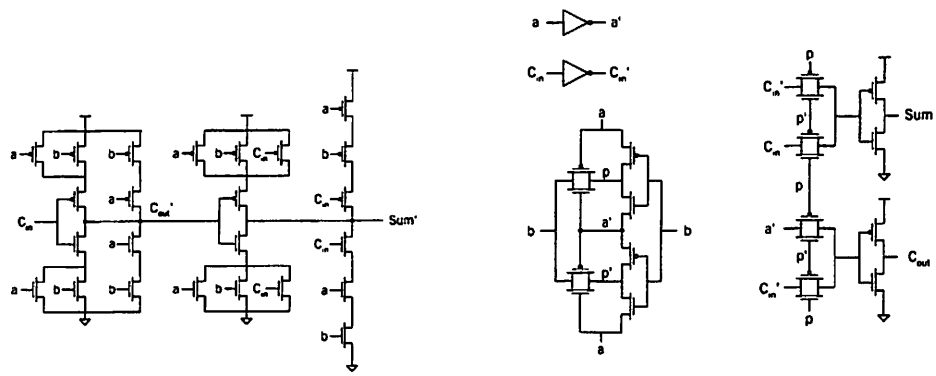
(b) Static passgate

(c) Dynamic (*np*-CMOS)

Figure 2.3: Five stage NAND chain implementations.

= 10fF, to represent more realistic fanouts for datapath circuits. The NAND chain is implemented in static CMOS, passgate, and a dynamic *np*-CMOS domino topology, while the 4-bit adder is arranged in a mirror configuration [17] and designed in static CMOS and passgate. In all adder circuit schematics, a and b are the two input bits, while p represents the propagate signal ($p = a \cdot b$).

Because circuit delay is dependent upon the size of its output capacitive load, the raw delay and energy dissipation values of the inverter chain are designed to be much greater than the corresponding performance of the NAND chain or adder. As previously mentioned, the focus of this study is not on absolute magnitudes of these delays, but rather the statistical spread of values. Therefore, the delay variability of



(a) Static CMOS (mirror configuration)

(b) Static passgate

Figure 2.4: Four-bit adder implementations.

all designs is calculated as the 1-σ standard deviation of the values normalized to the mean delay ($\frac{\sigma}{\mu}$), for all analyses.

2.4 Statistical Analysis Model

The delay measurements from Monte Carlo simulations produce a range of performance abilities for each circuit. The robustness of each design is evaluated by fitting a statistical distribution curve to the data and extracting the mean and variance values. Details of the statistical curve fitting and its accuracy are now discussed.

2.4.1 Lognormal Distribution

A random variable X has a lognormal distribution if its natural logarithm $Y = ln(X)$ has a normal (gaussian) distribution. The mean μ and variance σ^2 of the lognormal distribution are defined in terms of the mean m and variance s^2 of the normally

distributed natural logarithm of X [18]:

$$m = \text{mean}[\ln(X)] \tag{2.7}$$

$$s = \text{stdev}[\ln(X)] \tag{2.8}$$

$$\mu = \exp\left[\frac{(2m + s^2)}{2}\right] \tag{2.9}$$

$$\sigma = \sqrt{\exp(2m + 2s^2) - \exp(2m + s^2)} \tag{2.10}$$

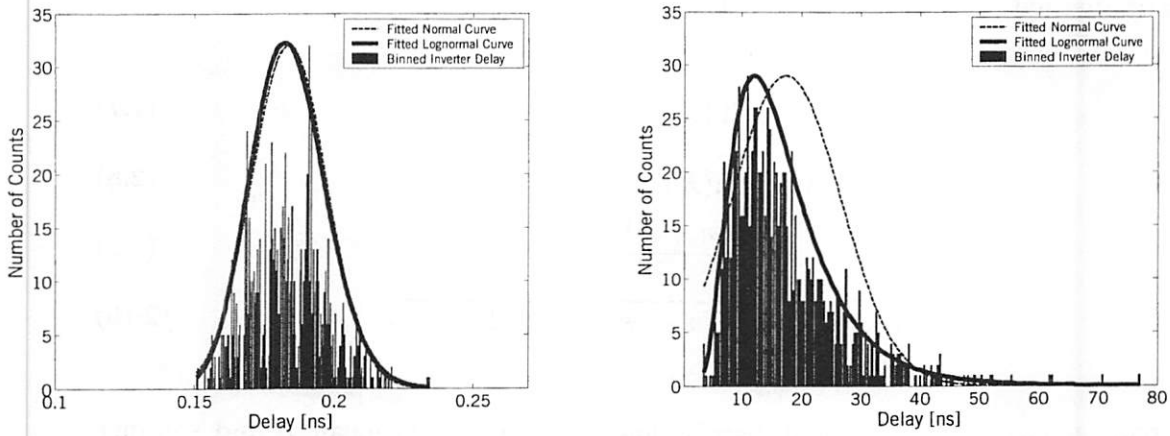
The shape of any lognormal distribution is defined by its mean μ and variance σ^2; these values are extracted from the delay data and used to predict variability in performance.

2.4.2 Least Sum of Squares Error

Because a fitted statistical distribution is used to model the behavior of the circuits under study, it is crucial that the error between the fitted curve and the simulated data be minimized. One method for measuring how well an estimated curve fits a set of data is using a least sum of squares error (SSE) calculation:

$$\text{SSE} = \sum_{i=1}^{n} (d_i - \hat{d}_i)^2 \tag{2.11}$$

where d is the actual data and $\hat{d}$ is the fitted data. This metric is useful for comparing between two or more fitted distributions to a set of data; the curve with the smallest SSE is determined to be the closest fit. However, the raw SSE value reveals minimal insight into the absolute accuracy of a fitted curve because its units are arbitrary and data-dependent. Therefore, the SSE metric is used as an estimate of the relative error of a fitted curve; a calculation of absolute error requires an additional metric, which is presented in Section 2.4.4.



(a) Nominal $V_{dd} = 1.2V$

(b) Lowered $V_{dd} = 300mV$

Figure 2.5: Comparison of fitting normal and lognormal curves to delay distribution of inverter chain.

2.4.3 Fitted Distribution Curves

Using Equations (2.9) and (2.10), the lognormal mean μ and variance σ^2 are extracted from each 1000-point delay distribution, and used to fit a curve to the data. Figure 2.5 compares the accuracies of the fitted lognormal and normal curves to the histogram of the inverter delay data under nominal and lowered V_{dd} conditions, with nominal $V_{th} = 240mV$.

Figure 2.5(a) illustrates the nearly overlapping fitted curves to the data when operating under nominal V_{dd}. Equation (2.11) is used to evaluate the respective sum of squares errors of each fitting to the data; the resulting error values match one another within 1%. This result indicates that either curve may be used to model the data with approximately equal accuracy for this case.

The accuracy of the lognormal fit to the data becomes more apparent in the lowered V_{dd} case, in which the distribution is heavily skewed to the left, with a long trailing tail toward larger delay values. Figure 2.5(b) illustrates the close fit of the

lognormal distribution to this set of data, which has a 30% smaller SSE compared with the normal distribution curve. Thus, the lognormal curve is shown to be the best fit to the data under all voltage conditions.

Although the SSE metric predicts the fitted shape of a data sample with high accuracy, its estimate of the absolute error tolerance of the fit is pessimistic. This is because the least SSE technique compares discrepancies between two arbitrary curves and sums error magnitudes, resulting in an increasing error estimate with increasing delay values. In contrast, the error between a set of data and its fitted probability density distribution (PDF) should decrease as delay increases. By definition, the higher the delay value is along the x-axis, the closer the area under the curve will tend toward the total number of samples, and hence the smaller the cumulative error should be. Thus, errors of opposite magnitudes should in fact cancel because points along the fitted distribution curve that overestimate the data count are compensated by those that underestimate the value.

A performance-based yield definition, based upon the lognormal delay distribution, is introduced in the next section and allows for the measurement of absolute fitting accuracy.

2.4.4 Performance-Based Yield Definition

A yield metric is defined with reference to delay results of the static inverter chain under nominal voltage conditions. Figure 2.6 replots Figure 2.5(a) with the histogram of delay values fit to a lognormal curve. The mean is $\mu = 184$ps, with a standard deviation of $\sigma = 13$ps, resulting in a normalized delay variability of $\frac{\sigma}{\mu} = 7\%$. The shaded area under the curve represents 95% of the total delay points, with a corresponding delay cutoff of 206ps, a value 12% greater than the fitted lognormal mean.

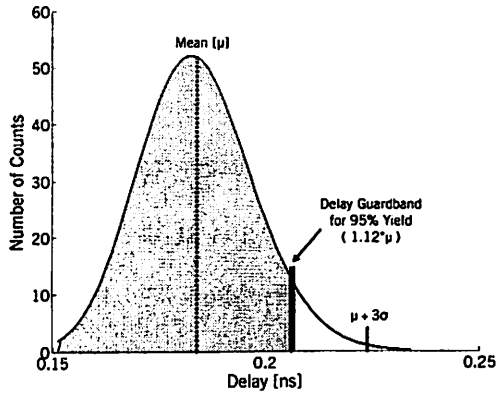


Figure 2.6: Performance-based yield definition based upon inverter chain under nominal conditions (V_{dd} = 1.2V, V_{th} = 240mV).

Analysis of this nominal case leads to the following yield definition:

$$\begin{aligned} \text{Delay Guardband} &\triangleq 1.12\times\text{fitted lognormal mean } (\mu) \\ \text{Yield} &= \%\text{ points falling within Delay Guardband} \\ &= \text{Probability } (\tau_d \leq 1.12 \cdot \mu) \end{aligned} \tag{2.12}$$

This fixed 12% guardband is chosen as the fixed delay cutoff point against which all circuits are compared, under all voltage conditions. According to this definition, the yield remains unchanged if the normalized delay variability $\frac{\sigma}{\mu}$ does not change, even if the raw μ and σ values do. The ability to maintain high yields in a design when subjected to increased parameter variations is an indication of circuit robustness; this metric quantifies the intrinsic level of performance variation control for each circuit.

Figure 2.7 replots Figure 2.5(b) with fitted lognormal mean and sigma points, which are used to calculate the yield reduction as a result of operating under aggressively lowered V_{dd}. In this case, not only are the mean and sigma delay values

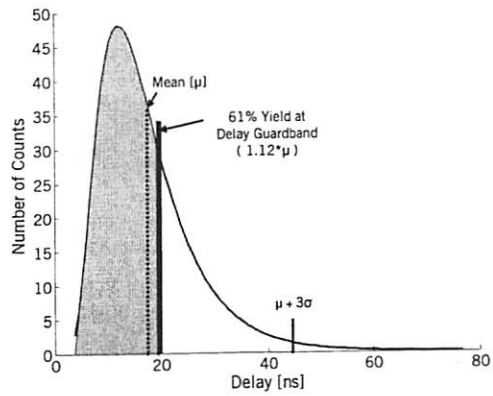


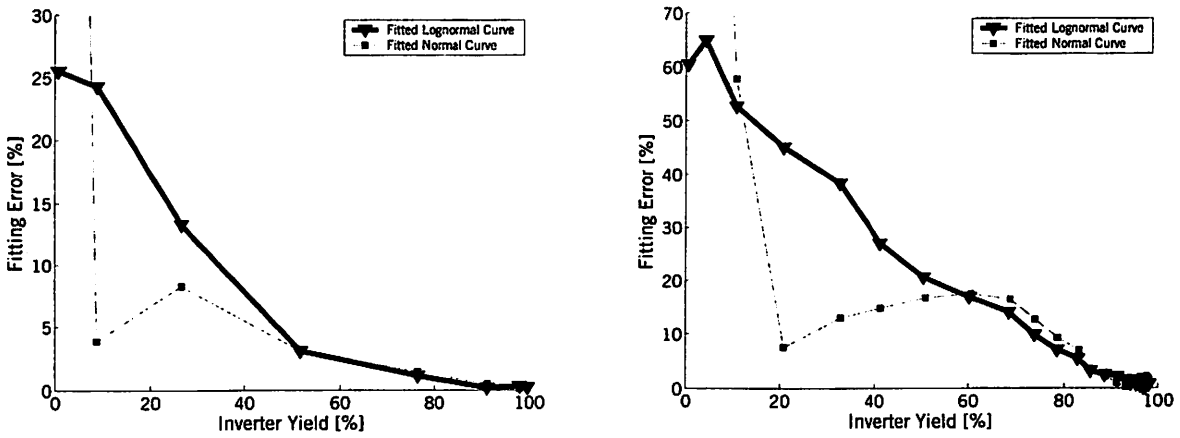
Figure 2.7: Yield of inverter chain under lowered V_{dd} (V_{dd} = 300mV, V_{th} = 240mV).

two orders of magnitude higher than in the previous case (μ = 17ns, σ = 9ns), the normalized delay variability $\frac{\sigma}{\mu}$ has increased by nearly an order of magnitude as well (from 6% to 53%). The implication of this more highly skewed distribution is that the resulting yield at the $1.12 \cdot \mu$ delay guardband has deteriorated to just 61%. Methods for compensating this loss will be investigated in Section 2.5.

An absolute measure for fitting accuracy is now explored for a range of expected and actual yield values. In all cases, the actual yield y is calculated by counting the number of delay values (out of the total 1000-point sample size) not exceeding the $1.12 \cdot \mu$ guardband. Meanwhile, the expected value $\hat{y}$ is predicted by the cumulative density function (CDF) of the delay guardband point, based on extracted mean and variance values. The fitting error is calculated as:

$$\text{Fitting Error [\%]} = \frac{|y - \hat{y}|}{y} \times 100 \tag{2.13}$$

Figure 2.8(a) plots this fitting error for the lognormal and normal distributions for nominal V_{dd}; Figure 2.8(b) plots an analogous plot for lowered V_{dd}.



(a) Nominal V_{dd}

(b) Lowered V_{dd}

Figure 2.8: Fitting error between inverter data and normal and lognormal curves.

These plots formalize the result shown in Figure 2.5; the shape of the delay distribution is best fit to a lognormal curve for all voltage conditions. The lognormal fitting curve exhibits a monotonically decreasing error, which indicates that it runs approximately parallel to the envelope of the delay histogram. In contrast, the fitted normal curve actually intersects the delay data, which is captured in the plot by a decreasing error magnitude that reaches a minimum and then sharply increases at the intersection point.

For yields above 80%, the accuracies of both fitted curves converge to an error tolerance within 5%. This implies that either fitting may be used to estimate high values with nearly equal accuracy; however the lognormal distribution is the better fit because it exhibits a slightly smaller absolute error and it captures the correct shape of the data across its entire range.

Note that because high yields may be predicted within a very high accuracy, the nominal 95% point is a somewhat arbitrary choice; any value above 90% may have

been chosen to serve as the standard for comparison.

An important result of the lognormal shape of the delay distribution is the implication that the circuit delay sensitivity to parameter fluctuations is nonlinear. Each of the parameter values is chosen randomly from a normal distribution; if they were to combine linearly, the output distribution would be normal as well. The fact that they actually combine nonlinearly indicates a higher-order interaction between parameter variation and delay variability. This complex interaction further motivates the derivation of an accurate, closed form expression for delay sensitivity to process parameters, so that performance metrics for future generation nanometer designs may be predicted without the dependence upon exhaustive Monte Carlo analyses.

The lognormally distributed delay model used to analyze both cases of V_{dd} for the inverter chain also fits the delay histograms for all other circuits in this study. The yield metric defined in this section is thus used to analyze all resulting data.

2.5 Simulations and Results

A known method for trading speed for low power in a design is by reducing V_{th} along with V_{dd} to recover performance loss [19]. In this work, all V_{th} reduction is assumed to be achieved through static design techniques (e.g. by specifying the channel doping concentration), rather through dynamic control of the body bias. The impact of this static V_{th} reduction is now explored for circuits experiencing parameter variability.

2.5.1 V_{dd} and V_{th} Optimization

The three-dimensional surface plots in Figure 2.9 show trends in delay, normalized delay variability ($\frac{\sigma}{\mu}$), switching energy, and leakage energy as functions of supply and threshold voltages, as simulated for the static CMOS inverter chain. As V_{dd} is scaled

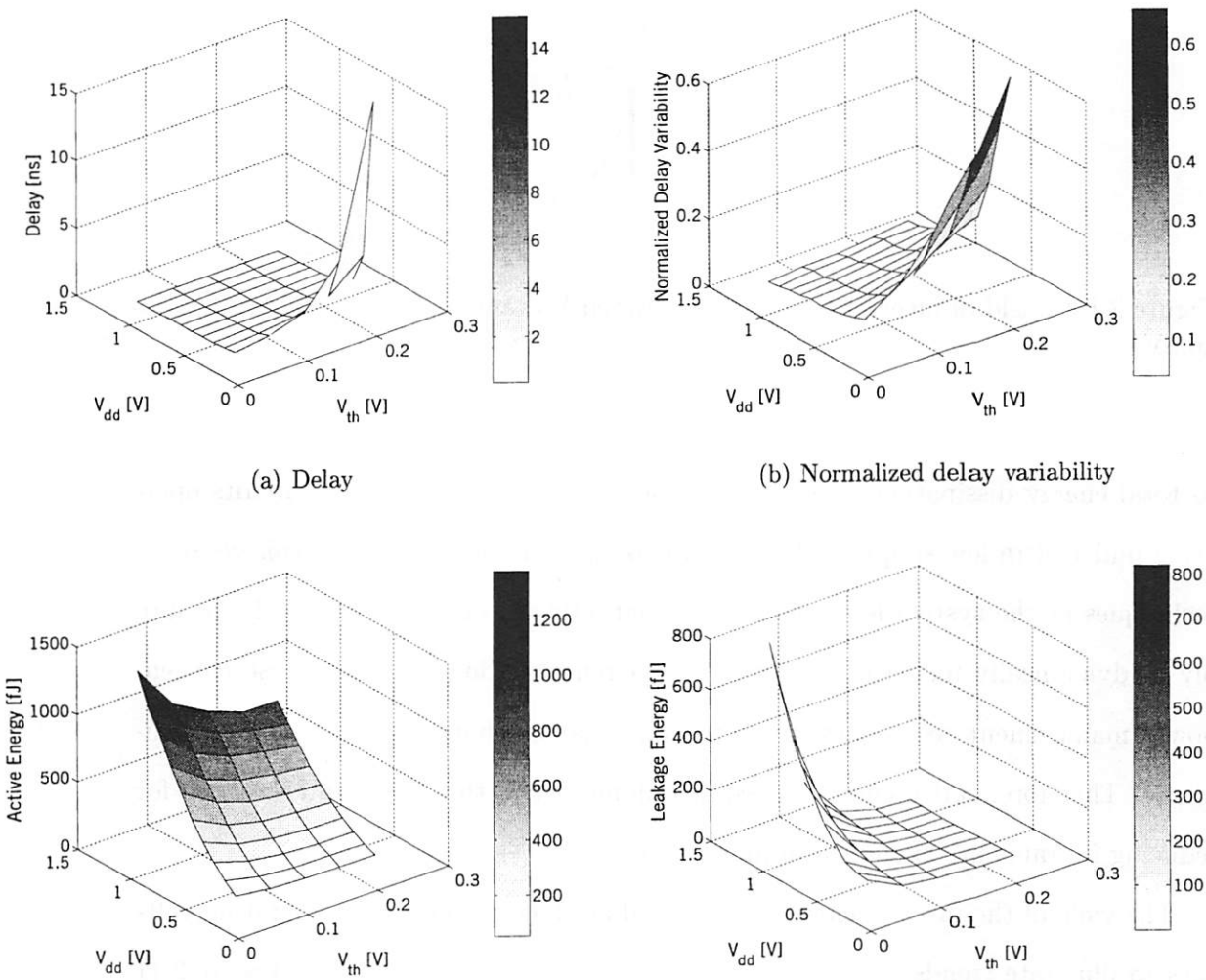
Table 2.2: Summary of inverter chain performance for nominal and reduced voltages.

V_{dd}	V_{th}	Mean Delay [μ]	Std Dev of Mean [σ]	Delay Variability [$\frac{\sigma}{\mu}$]	Yield	Active Energy
1.2V	240mV	184ps	13ps	7%	95%	798nJ
300mV	240mV	17ns	9ns	53%	61%	53nJ
300mV	40mV	385ps	33ps	9%	92%	114nJ

from its nominal value of 1.2V to a reduced 300mV, while V_{th} is maintained at its nominal value, active energy dissipation is decreased by over an order of magnitude (Figure 2.9(c)), but both delay and the spread of delay have increased by two orders of magnitude (Figures 2.9(a), 2.9(b)). Clearly, the dramatic increase in delay and delay variability outweigh the benefits associated with the energy savings in this scenario, and further reduction in V_{th} from its nominal 240mV to 40mV is necessary to restore performance to a desirable level.

The improvement gained by scaling both V_{th} and V_{dd} together is revealed by fitting a lognormal curve to the inverter delay data at this voltage combination and calculating the yield, as discussed in Section 2.4.4. Figure 2.10 illustrates the improvement in delay variability with this V_{th} scaling: the mean and standard deviation values for delay have been restored closer to their nominal values (μ = 385ps, σ = 33ps) resulting in a 92% yield. Table 2.2 summarizes delay, delay variability, yield and active energy for nominal and lowered voltage conditions.

One significant disadvantage of operating at this low V_{dd} and V_{th} point is the resulting increase in leakage energy. The magnitude of this leakage energy is dependent upon many factors, including the average gate activity factor (α), transistor stack height, and process effects (e.g. subthreshold slope and DIBL). The leakage energy shown in Figure 2.9(d) is simulated with an activity factor $\alpha = 0.5$. Under the condition of lowered V_{th}, this leakage component threatens to contribute significantly



(a) Delay

(b) Normalized delay variability

(c) Active energy

(d) Leakage energy

Figure 2.9: Surface plots of performance for an inverter chain across the (V_{dd},V_{th}) design space.

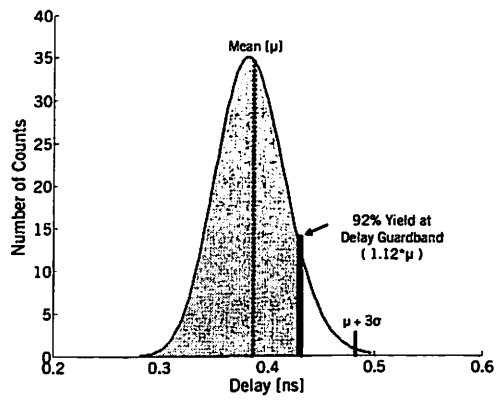


Figure 2.10: Yield of inverter chain under lowered V_{dd} and V_{th} (V_{dd} = 300mV, V_{th} = 40mV).

to total energy dissipation. However, it is assumed in this work that circuits operating under ultra low supply and threshold voltages are designed with leakage-aware techniques at the system level, such as disconnecting circuits from the voltage supply or dynamically increasing V_{th} when the circuit is idle [4]. Using these efficient power management techniques in sleep mode ensures that leakage energy is minimized. Therefore, active energy is assumed dominant in this work, and methods for reducing its value are examined in more detail.

The yield of the inverter chain is calculated at all combinations of operating voltages to illustrate trends with V_{dd} and V_{th}. These results are plotted in Figure 2.11 and show that selecting a lower threshold voltage along with supply voltage improves both delay and delay variability. For example, though the yield falls from 95% to 60% when V_{dd} is reduced from 1.2V to 300mV and V_{th} is held at its nominal value, the technique of also lowering V_{th} (to 40mV) restores this loss to over 90%.

The three-dimensional surface plots for the inverter chain are representative of analogous plots for all other circuits under study.

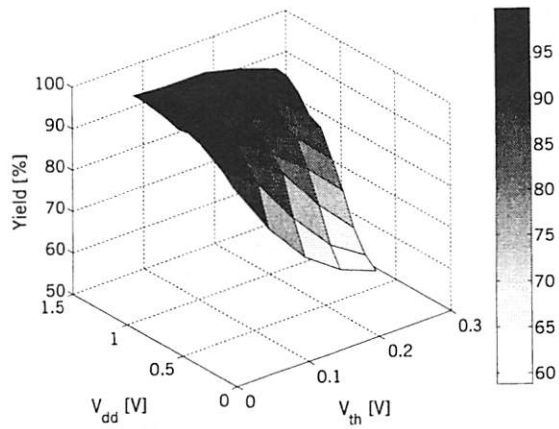
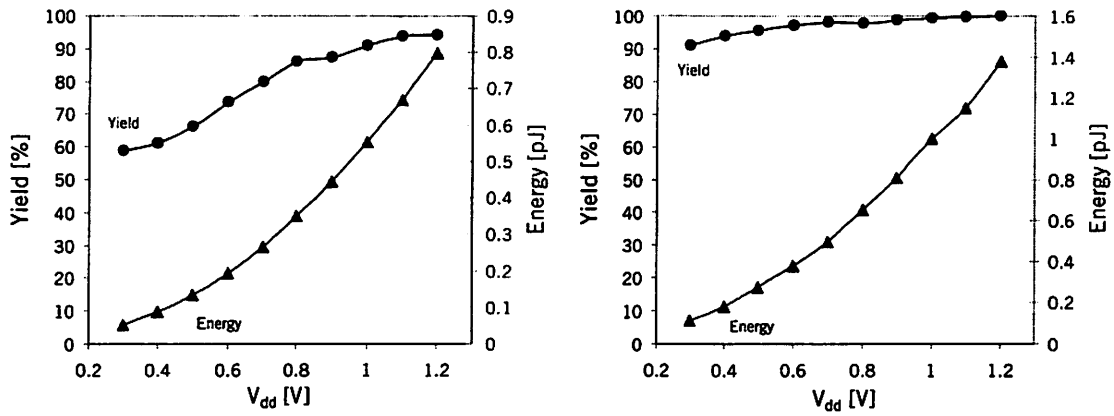


Figure 2.11: Circuit level yield dependence on V_{dd},V_{th} of an inverter chain.

2.5.2 Yield-Energy Tradeoffs

Results from the previous section showed that reduction of V_{th} with V_{dd} is an effective method for mitigating performance variability. Tradeoffs between yield and energy are investigated by fixing V_{th} at its nominal and lowest design values, and generating two-dimensional contour plots from the simulated inverter data at these two threshold voltages. Figure 2.12(a) motivates the need for compensating yield degradation; as V_{dd} is scaled from 1.2V to 300mV while maintaining V_{th} at 240mV, active energy dissipation is reduced by 93%, but the corresponding yield plummets by 38%. In comparison, the tradeoff in Figure 2.12(b) is more favorable because V_{th} is scaled along with V_{dd} (to 40mV in this case), and reduces the yield loss to just 9% while still benefitting from a high energy savings of 92%.

These results show that the values of circuit operating parameters V_{dd} and V_{th} may be chosen such that a small sacrifice in performance gains a significant reduction in energy consumption, despite increased processing variations and the resulting increased delay variability. In the case of the inverter chain, simultaneously lowering



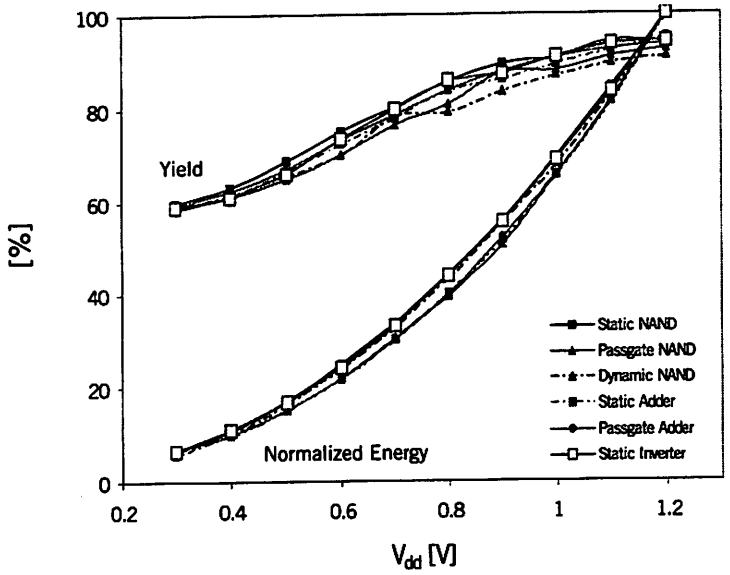
(a) V_{th} = 240mV (b) V_{th} = 40mV

Figure 2.12: Yield-energy tradeoffs for inverter chain at nominal and reduced V_{th}.

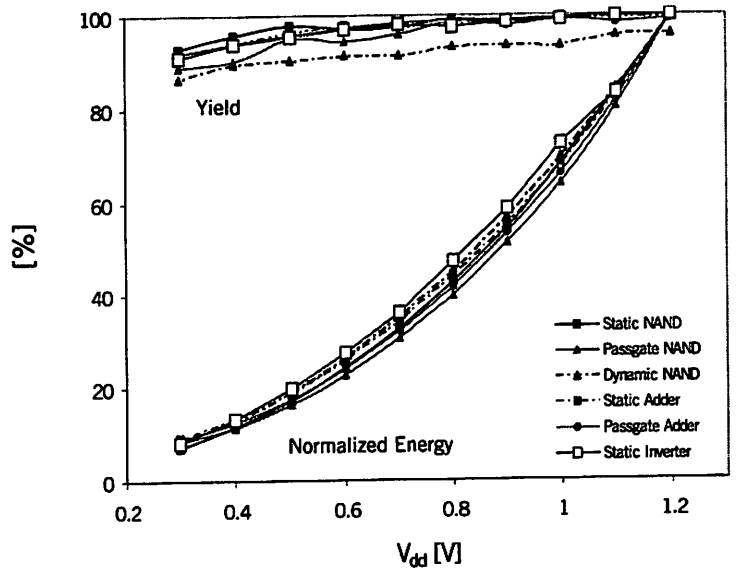
V_{th} with V_{dd} reduces active energy consumption by over 90%, for a comparably slight 9% reduction in yield. This tradeoff is favorable for ultra low power design; if errors resulting from the 9% of circuits failing to meet timing specifications are compensated using low overhead fault tolerant techniques, then the circuit may still benefit from significant energy savings.

Trends in the yield-energy tradeoff for more complex circuits using various topologies agree with results for the inverter chain. Figure 2.13 plots yield and energy contours for all circuits studied, for both V_{th} values. The energy dissipation curves are normalized in order to compare the savings in active energy for all circuits on the same plot.

Figure 2.13(a) shows that in the nominal V_{th} case, all circuits suffer from significant performance variability, even at the nominal V_{dd} point for which the average value for all circuits is 93%. Furthermore, as V_{dd} is scaled, this average drops to 59%, indicating a 37% loss. Meanwhile, the average energy savings from nominal to reduced V_{dd} is 93%. Thus at nominal V_{th}, an approximate 40% sacrifice in yield leads



(a) V_{th} = 240mV



(b) V_{th} = 40mV

Figure 2.13: Yield degradation trends for all circuits and topologies evaluated.

to an energy reduction of over an order of magnitude. This trend is fairly constant across circuit complexity, with small differences between logic evaluation styles. The static CMOS family emerges as the most robust, displaying the overall highest yield percentages, with the passgate designs slightly more vulnerable. The circuit topology with the most variation is the dynamic implementation of the NAND chain, whose performance is overall the least robust to process variations.

Figure 2.13(b) illustrates the effectiveness of recovering yield using V_{th} scaling; a 99% average is achieved at nominal V_{dd}, dropping to only 91% as V_{dd} is lowered. This 8% sacrifice produces average energy savings of 92%, which is a much more favorable tradeoff than in the previous case. Furthermore, this trend is consistent across circuit complexity, with a slight dependence upon logic evaluation style. The robustness of the static CMOS topology remains the highest, while the passgate implementations of the NAND and adder are slightly less robust. Meanwhile, the discrepancy between the dynamic NAND chain and other circuits is even more pronounced in this reduced V_{th} case, suggesting that dynamic logic families are inherently more susceptible to variations in process and operating parameters. However, the divergence of dynamic NAND results from other cases is maintained within 4%, indicating that the overall results are consistent.

2.6 Discussion

Results from Monte Carlo simulations indicate that aggressive V_{dd} scaling resulted in significant energy savings, but at the cost of prohibitively increased delay magnitudes, as well as the spread of these values. A yield metric is defined to quantify the amount of delay variability experienced by an arbitrary circuit, and thus serves as a measure of robustness to variations. Tradeoffs between active energy dissipation and this

performance-based yield are investigated for various circuits under a range of voltage conditions.

The most effective method for improving circuit performance while maintaining benefits from significant active energy reduction is the technique of reducing V_{th} simultaneously with V_{dd}, not necessarily keeping the ratio $\frac{V_{dd}}{V_{th}}$ constant. As V_{dd} is reduced from 1.2V to 300mV for the inverter chain, a corresponding decrease in V_{th} from 240mV to 40mV is shown to lower energy dissipation by 92%, with a comparatively minor 8% yield loss. Moreover, these results, along with overall trends in delay, delay variability, and energy across the V_{dd},V_{th} design space, are consistent for the circuit types and implementing topologies studied. The family of static CMOS circuits is determined to be most robust, while the dynamic implementation of the NAND chain suffered the highest vulnerability to process variations. However, total yield loss for all circuits studied matches within 4%, indicating a comparable response in performance to manufacturing and operational variations for all designs. A larger, more complex family of circuits, using static CMOS, passgate and dynamic styles, will be studied in Chapter 3.

These results motivate novel approaches for fault tolerant design, in order to correct for the average remaining 9% of circuits that fail to meet the timing specification. A system level fault tolerant scheme for correcting circuit level timing errors is investigated and discussed in Chapter 4.

2.6.1 Future Work

There are a number of aspects of the experimental setup used in this study that may benefit from further research.

First, L_{eff} was assumed to be the most dominant physical parameter affecting

V_{th} variations, and hence, delay variability. However, the dependence of $\sigma_{V_{th}}$ on W is nonzero, as shown in Equation (2.5), and may be significant. Preliminary studies indicate that the accuracy of yield estimates may be improved by over 5% as a result of including fluctuations in transistor widths [20].

In this work, the values chosen for all variable parameters were assigned uniformly to all transistors within a simulation. As previously discussed, this is not an accurate assumption for V_{th} variations, which experience random dopant fluctuations independently of physical proximity to other devices. Furthermore, perfect spatial correlation will not hold for designs of larger scales; transistors will inevitably experience mismatch in channel lengths, especially as logic depths increase and the physical distance between devices widens. Efforts are presently underway to investigate spatial correlation levels in transistor gate lengths, using exhaustive measurements of critical dimensions from a full 200mm wafer processed through a standard 130nm manufacturing process. Results indicate a high spatial correlation ($0.8 \leq \rho \leq 1$) in gate length for transistors separated by vertical and horizontal distances of up to 2mm [21]; further work is needed to refine this model.

As previously mentioned, techniques of containing leakage energy within limits of the power budget is crucial for the success of low power designs, in which gate lengths and supply voltage continue to shrink. Techniques of applying forward body biases to leaky transistors and shutting off the supply to circuits during sleep mode are only a few of the ideas that are currently under research.

Finally, although the Monte Carlo experimental setup was useful for predicting the complex interaction between process variations and performance variability, the results were based upon more than 600,000 SPICE simulations completed over a number of weeks. This exhaustive methodology is clearly not reasonable for use in studying variabilities in complete VLSI designs; a more simplified and efficient analy-

sis methodology is needed. Results from this work indicate that the delay variability and energy consumption of the static inverter chain across the V_{dd}, V_{th} space is representative of similar trends in more complex circuits, such as NAND chains and adders, as well as those implemented in various other logic styles. This suggests that an arbitrarily complex circuit may be modeled by a more simple network of circuits (e.g. static inverter chains), in order to gain insights into its expected yield and energy tradeoffs. However, this simplification cannot be made until the above assumptions are verified.

Chapter 3

Process/Circuits Co-Design: Power and Delay Variability

Preliminary studies in Chapter 2 discussed the critical importance of accurately predicting performance metrics such as energy and timing margins for successful designs in current and future technologies. However, while the previous study focused on aggressive scaling trends in bulk CMOS, fundamental limits to conventional device scaling techniques loom in the foreseeable future [22], in most part due to critical linewidths approaching atomic dimensions. Responding to pessimistic forecasts, current directions for technology evolution rely increasingly on process and device innovation, rather than on the steadfast shrinking of physical transistor dimensions.

For example, VLSI industry leaders have pursued and refined the technology of partially depleted silicon-on-insulator (pd-SOI) in recent years, a technique that insulates each active transistor from the silicon substrate with a thick insulating layer of silicon dioxide (SiO_2). When compared to conventional bulk CMOS technologies, pd-SOI benefits from three key features intrinsic to its physical structure [23]:

- *Reduced junction capacitances.* The source and drain regions of an SOI device border the insulating oxide layer rather than the silicon substrate. The dielectric constant (K) of SiO_2 is approximately three times smaller than that of pure

silicon, resulting in significantly reduced parasitic junction capacitances.

- *Reduced short channel effect.* As the drain voltage increases, charge from the drain leaks into the floating body, in turn increasing the body potential. As the body potential rises, the space-charge region between the drain and body shrinks, thereby increasing effective channel lengths. This effect mitigates the rapid roll-off of threshold voltage with short channels, a challenge characteristic of bulk designs.

- *Reduced average device threshold voltages due to floating body bias.* As the body potential increases with drain potential, the junction diode between the bulk and source becomes slightly forward-biased. Due to the body effect, the effective, operating threshold voltage decreases from its intrinsic value, boosting the performance capability of the pd-SOI design.

These three characteristics of pd-SOI circuits combine to improve circuit performance, compared to conventional designs in bulk technologies. Unfortunately, these advantages come with significant challenges: when creating analytical models to predict pd-SOI circuit behavior, traditional scaling theories based upon bulk designs are insufficient and may not be applied, because the underlying physics of pd-SOI differ from its bulk predecessor. Furthermore, pd-SOI technology also faces increases in process variations as the state of the art scales into the nanometer regime. As manufacturing and process variations become harder to control, the ability to accurately predict critical performance metrics decreases as well. In order to evaluate the response of pd-SOI circuits to variations in process and operating parameters, a Monte Carlo simulation-based research effort similar to the one described in Chapter 2 is pursued at an industrial research site using 90nm pd-SOI technology.

The overall goal of this study is to determine the delay and power variability for a set of representative circuits whose parameters are subjected to manufacturing and operating fluctuations. First, the effects of all parameters varying simultaneously are studied in order to simulate realistic spreads in delay and active power. Next, the parameters are isolated and varied individually, in order to quantify the contributions from each toward these total variabilities. Further details of the experimental setup are discussed in the next section, followed by the presentation and discussion of simulation results in Section 3.2. Directions for future work in Section 3.3 conclude this chapter.

3.1 Experimental Setup

The experimental setup for the industrial research study is described in two parts: the Monte Carlo simulation framework and design of the circuits submitted to the simulations.

3.1.1 Monte Carlo Simulation Framework

Although the logistical framework of this Monte Carlo simulation is similar to the one shown and described by Figure 2.1, there are a few differences in the parameter setup process due to studying the performance variabilities with a manufacturing and process emphasis rather than from a voltage optimization perspective.

First, because the focus of this study is on investigating the sensitivity levels of process parameters rather than in tuning operating parameters to find an optimal design corner, there is no effort to aggressively scale and discretize the range of nominal V_{dd} and V_{th} values. Instead, the number of physical process parameters varied is increased to include W and t_{ox}, and the parameter L represents the drawn polysilicon

gate length, rather than the effective channel length. Finally, because the goals of this study required gathering two different sets of data, two Monte Carlo simulations are performed for each circuit in this study. The details and differences between the two frameworks are now described.

Monte Carlo Simulation I: All Parameters Varying

Each Monte Carlo simulation consists of a batch of SPICE simulations of a given circuit. The number N of SPICE simulations per batch is dependent upon the circuit complexity; circuits comprising relatively more transistors are submitted to a fewer number of simulations because each requires significantly more time to complete. For each SPICE simulation, all circuit parameters are drawn randomly from their respective distributions, with all interactions between variable parameters (e.g. V_{th} dependence upon W, L and t_{ox}) reconciled by the simulator, similar to the method described in Section 2.3.1. Distributions for the process parameters W, L, V_{th} and t_{ox} are specified by the BSIM SOI model [24], with limits consistent with those predicted by the ITRS [5]. The operating supply voltage V_{dd} is varied over a normal distribution with a nominal value of 1V and 3σ value of 50mV. The operating temperature is held at a maxiumum value of 85°C because even large temperature variations are found to have negligible effects on circuit performance variability.

The spatial correlation coefficient ρ is again set for perfect parameter correlation ($\rho = 1$) and held at this value for all simulations. Similarly to the study in Chapter 2, this assumption leads to a worst-case scenario for overall variability levels.

After values of all parameters are chosen, the circuit is submitted to a SPICE simulation and its active power dissipation and delay are measured. This process is then repeated; the next circuit instance is created by choosing and applying a new set of random parameter values, and the resulting netlist is analyzed with SPICE. Figure 3.1

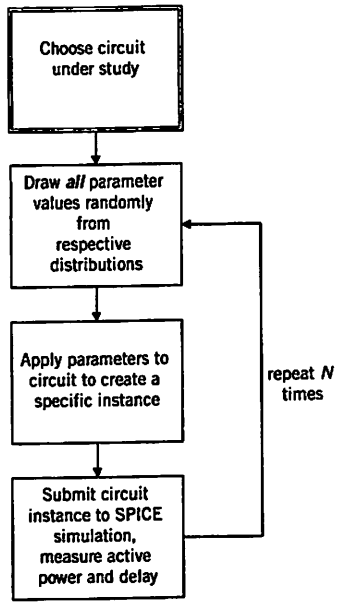


Figure 3.1: Monte Carlo simulation I: All parameters varying.

diagrams the simulation framework allowing all parameters to vary simultaneously. After the entire batch of N circuit instances is submitted to SPICE simulations, a statistical analysis of the resulting performance metrics for that circuit is performed. This Monte Carlo simulation is then repeated for the next representative circuit.

Monte Carlo Simulation II: Individual Parameters Varying

While the purpose of the previous Monte Carlo simulation is to study variations in the active power and delay of a circuit, it does not provide insights into how each uncertain parameter affects overall variability levels. When considering the impact of manufacturing-induced variations on circuit design, it is important to identify the parameters affecting circuit performance variability the most significantly, so that the variation tolerance of the manufacturing process may be improved for that parameter. In this case, statistical methods for extracting the individual parameter contributions from overall variability levels do not accurately model the interactions

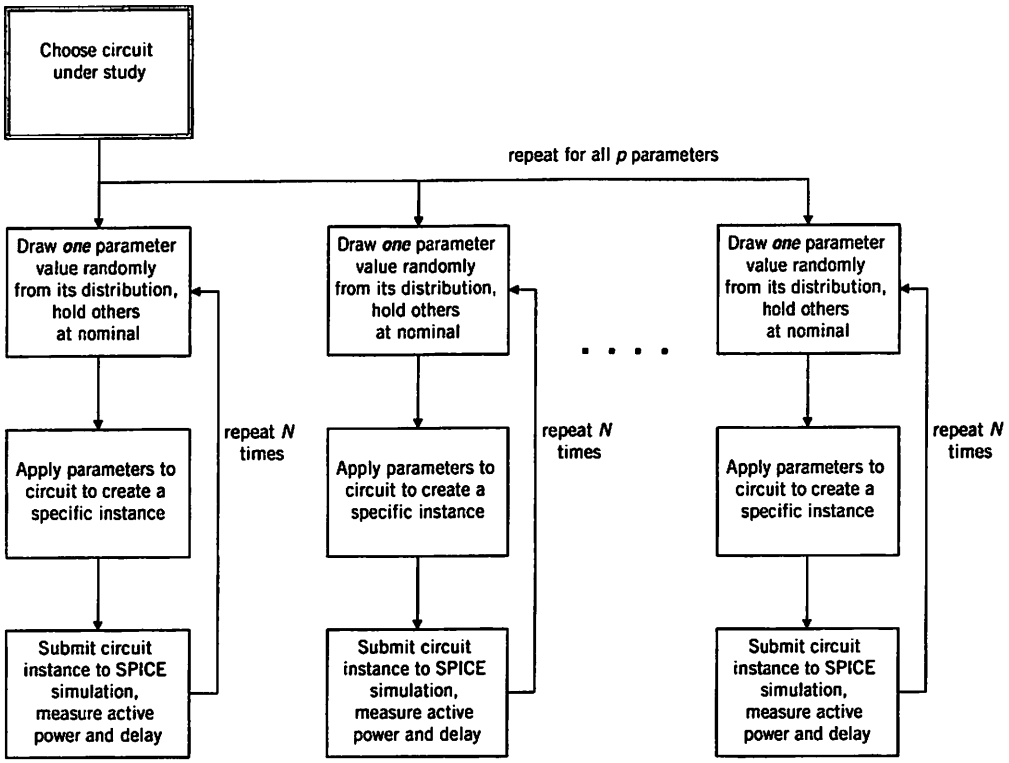


Figure 3.2: Monte Carlo simulation II: Individual parameters varying.

between parameters, namely the dependency of V_{th} on W, L and V_{th}. Therefore, each circuit under study is submitted to a second, independent Monte Carlo simulation, which decouples the contribution of each parameter to the overall variability. A block diagram for this simulation framework is shown in Figure 3.2.

The parameter distributions and operating temperature remain unchanged from the first Monte Carlo simulation setup. However, only one parameter value is drawn randomly from its distribution for each SPICE submission while all remaining parameters are held constant at their nominal values. When compared to the first simulation setup, the total number of SPICE simulations required for this methodology has increased by a factor of p, the number of parameters that vary.

3.1.2 Circuits Under Study

The circuits under study represent basic datapath elements of typical microprocessors. For the scope of this work, two circuit functions are chosen: a six stage chain of NAND gates and a family of 16-bit adders. Furthermore, each of these circuit types is designed using multiple logic evaluation styles and in the case of the adder family, different circuit architectures, so that a comparison may be performed between various implementations of the same logic function. The value of submitting multiple designs of the same circuit function is in providing insights into the degrees of inherent robustness of each to variations in manufactured parameters.

The remainder of this section is organized as follows. First, the designs of NAND chains are discussed, using circuit schematics to illustrate the various logic evaluation styles used. Then, the circuit complexity is increased and the family of 16-bit adders is described and shown using block diagram representations for each of the architectures implemented. Finally, after the two circuit types are introduced, the methodology used for optimizing all transistor sizes in all designs is discussed.

NAND Chains

Each of the NAND chains consists of a series of six three-input NAND gates, with intermediate outputs feeding into successive inputs. In total, five NAND chains are designed using three logic evaluation styles: static CMOS, dynamic, and passgate. In all cases, the critical path is excited by tying two of the three inputs high (to V_{dd}), while switching the third input from high to low and back high again. Furthermore, the switching input is systematically placed as far down along the transistor stack as possible, away from the output node, to stimulate the longest path from any input to the output. Because the number of stages along the chain is even, the final output

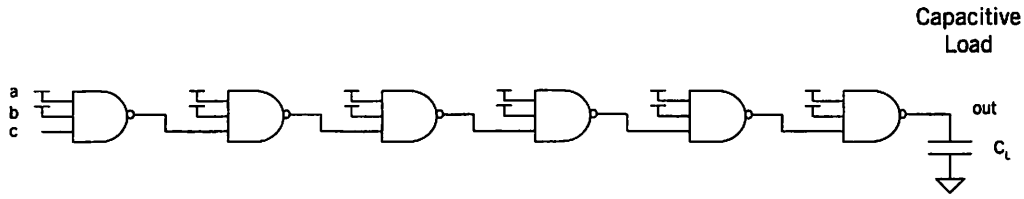


Figure 3.3: NAND chain with static capacitive loading.

transitions in the same direction as the input (both switch either high-to-low or low-to-high).

Output loading The output of the NAND chain is loaded with a static capacitor of value C_L = 10fF, consistent with the input capacitance of a typical successive stage. This loading scheme is shown in Figure 3.3, and is submitted to both types of Monte Carlo simulations. Because the static load is modeled as an ideal, passive capacitor in the SPICE simulation, its value remains constant throughout each of the simulations and is unaffected by the random parameter selection process. In order to compare this scheme with one that more realistically models the fluctuating input capacitance of an active successive stage, a second loading condition is designed using fanout-of-three (FO3) loading. The schematic in Figure 3.4 shows the six stage chain loaded with a FO3 load, which comprises three identical NAND gates, each implemented in the same logic evaluation style as all other NANDs in the design. Because this load contains active devices, each of its transistors is subjected to the same process parameter variations as the other gates that form the chain. This FO3 loaded NAND chain is submitted to the second Monte Carlo simulation described, in which individual parameters are isolated and varied individually. The resulting delay variability breakdown by parameter is then compared with results from the static capacitor loading case.

In order to disregard any unwanted slewing effects caused by the rise and fall times

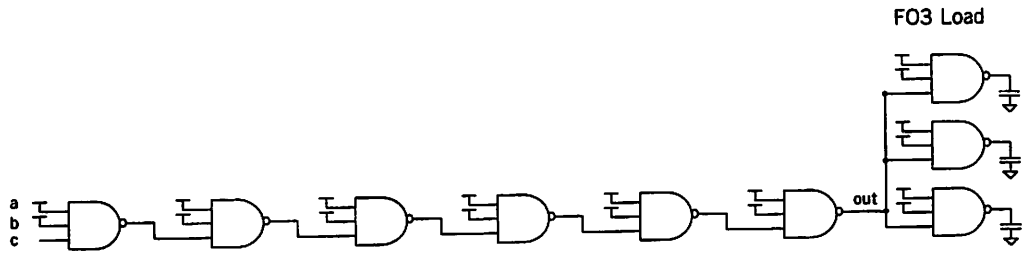
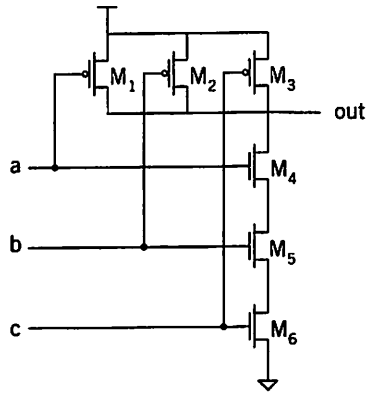


Figure 3.4: NAND chain with FO3 loading.

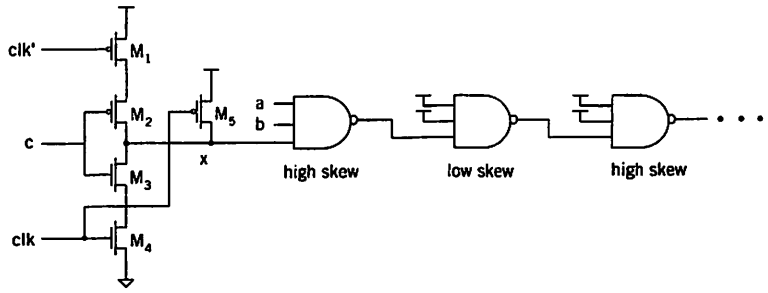
of the ideal input signal or by the output loading of the final stage, all propagation delays are measured from the input of the second stage to the output of the fifth stage. Each NAND chain is submitted to a total of $N = 1000$ SPICE simulations. The four different logic evaluation styles implemented are now discussed.

Static CMOS There are two implementations of the NAND chain based on the complementary static logic style; the first is designed in pure complementary static CMOS. Figure 3.5(a) on page 48 shows a standard circuit schematic for this first case: transistors $M_1 - M_3$ form the PMOS pull-up network while $M_4 - M_6$ form the NMOS pull-down chain.

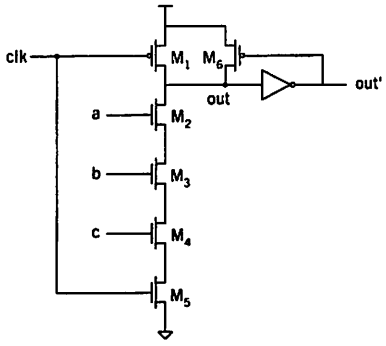
Pulsed Static Figure 3.5(b) illustrates a variant on the standard complementary static case. The third input signal (labeled `x`) is pre-charged to a known state at the start of each circuit evaluation and then fed into alternating high- and low-skewed successive stages that favor each monotonic transition. This technique is known as Pulsed Static CMOS (PSCMOS), and achieves improved performance over pure complementary static CMOS without the burden of the large clock load characteristic of pure dynamic styles [25]. In this implementation, PMOS transistor M_5 pre-charges node `x` when the clock signal `clk` is low, and turns off when `clk` switches high, allowing the circuit to evaluate.



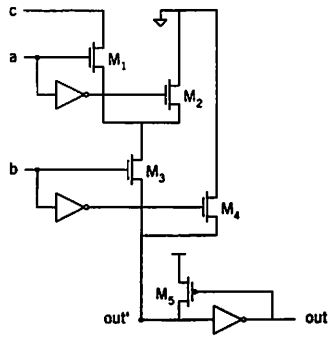
(a) Static CMOS



(b) Pulsed static CMOS



(c) Dynamic domino



(d) Static passgate

Figure 3.5: Three-input NAND gate implemented in various logic evaluation styles.

Dynamic A standard dynamic domino NAND implementation is shown in Figure 3.5(c), with transistors M_1 – M_5 forming the dynamically evaluated circuitry. The feedback network containing a small static inverter and minimally sized keeper transistor M_6 mitigate the effects of current leakage by reinforcing a high value on the dynamic node out when one or more inputs a, b, and c is low.

Passgate The schematic shown in Figure 3.5(d) uses a passgate-based logic evaluation style from the Lean Integration Using Passgates (LEAP) library, a set of cells with sufficient flexibility to realize a full set of logic blocks with minimal complexity, designed for use in automated design methodologies [26]. Because this implementation relies on NMOS transistors placed in series to couple input signals to outputs, each transistor experiences a voltage drop of V_{th} across its source and drain when passing a high logic level. Without additional level restoring circuitry, the resulting voltage at node out$^-$ may be as low as $2V_{th}$ below V_{dd}. The static inverter and small keeper transistor (M_6) are thus placed at node out$^-$ to recharge its value to the full V_{dd} rail when high values are passed.

16-bit Adders

In total, eleven 16-bit adders are designed and submitted to both Monte Carlo simulations, spanning a range of circuit architectures and logic evaluation styles. In contrast to the relatively simple NAND chain design, with just one output node, each adder has 32 outputs: 16 sum bits and 16 carry out bits. In most cases, the critical path is between the initial carry in bit (C_{in0}) and the sixteenth carry out bit (C_{out15}), with one exception in the case of the irregular Brent Kung carry lookahead tree configuration, in which the critical path output node is C_{out14}.

A fanout-of-four (FO4) static inverter is used to load the critical paths for all

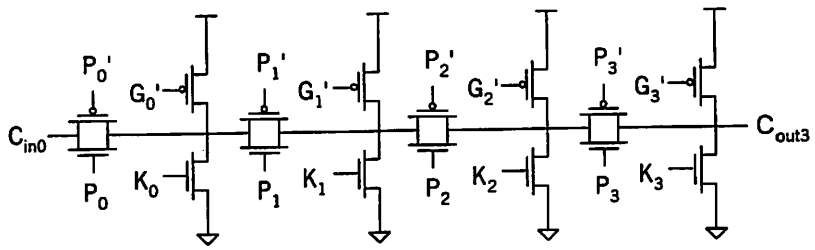
adder designs. Due to the increase in both logic complexity and transistor count as compared to the NAND chains, a reduced number of simulations ($N = 200$) is run.

Ripple Carry The most basic adder style studied is the ripple carry adder, whose 4-bit slice block diagram was presented in Figure 2.2(c) in Chapter 2. In this implementation, a passgate-based Manchester carry chain is used to quickly propagate the C_{in} bit to the output by means of a series of passgates connecting consecutive bits [27]. Four-bit slices of static and dynamic implementations of the Manchester carry chain are shown in figures 3.6(a) and 3.6(b), respectively. In order to complete the 16-bit configuration, the blocks are repeated four times and connected in series with static inverters inserted between each four bit block to recover full rail swings at intermediate nodes.

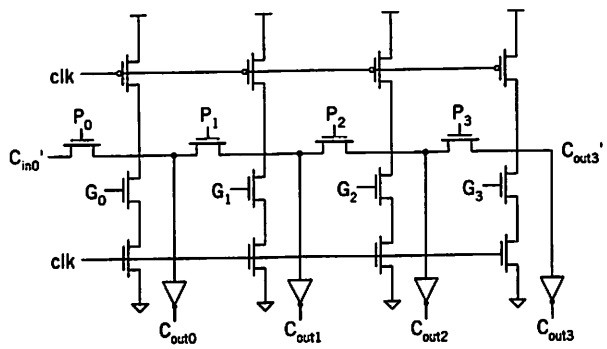
Carry Select Figure 3.7 shows a carry select adder arranged in a logarithmic configuration, which equalizes signal arriving times by incrementally increasing the lengths of successive blocks. The pre-computed sum and carry out bits of each stage arrive at approximately the same time as the multiplexor select signal from the previous stage, thereby reducing total delay when compared with a linear configuration [27]. The carry select adder is implemented in static CMOS, dynamic domino, and passgate styles.

Carry Lookahead Trees When compared with the basic ripple carry and carry select styles, the carry lookahead adder is the architecture of choice for high performance applications due to its superior speed, though at the cost of increased area and power consumption.

The carry lookahead tree with the most regular and full layout is the Kogge Stone configuration, which systematically combines consecutive bits using circuits that pro-



(a) Static implementation



(b) Dynamic implementation

Figure 3.6: Manchester carry chain for ripple carry adder (sum generation not shown).

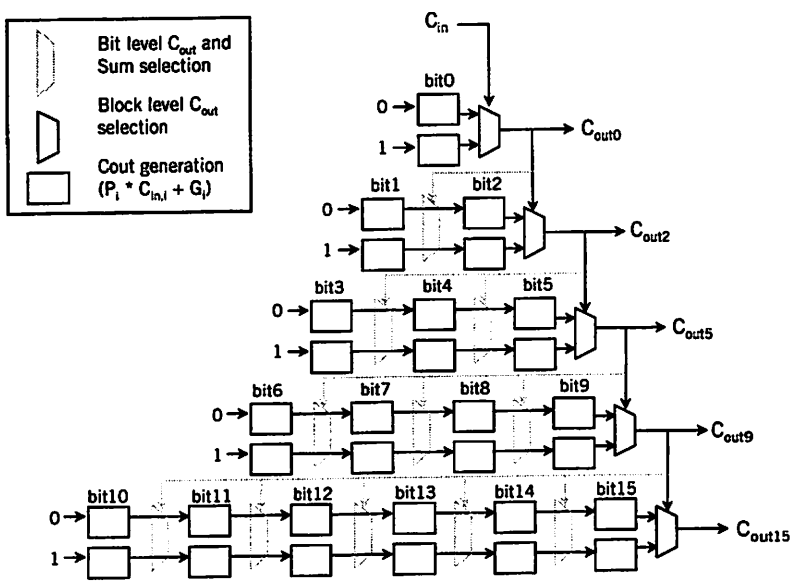


Figure 3.7: Sixteen-bit, logarithmic carry select adder.

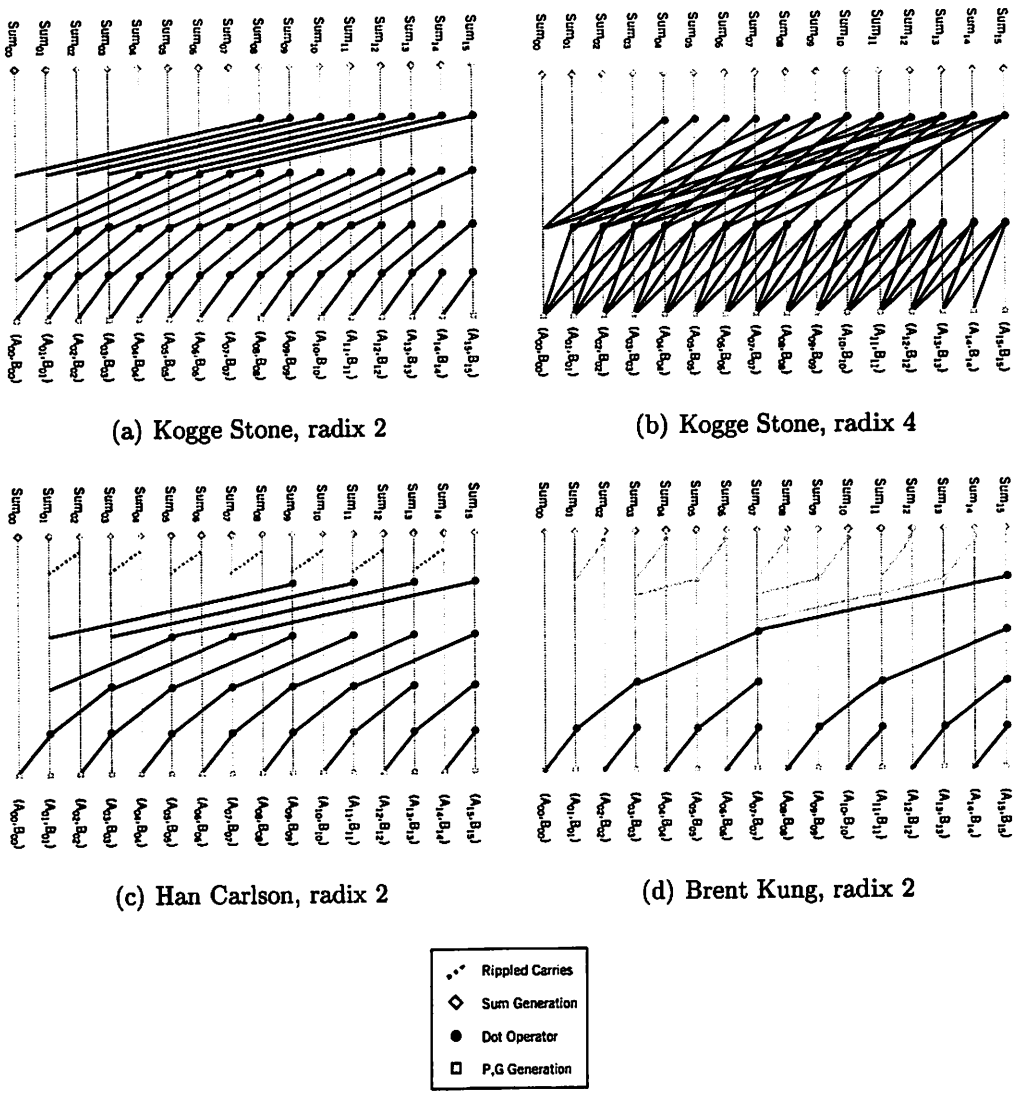
duce group propagate and generate signals (sometimes called *dot operators* [27]) along successive logic depths until the final carry out bit value is calculated [28]. The number of consecutive bits combined at each tree node is defined as the *radix*; a radix 2 Kogge Stone tree combines pairwise bits while a radix 4 Kogge Stone tree combines bits in groups of four. Figures 3.8(a) and 3.8(b) illustrate trees with these radix values.

A disadvantage of fully populated Kogge Stone trees is the large area penalty resulting from the circuitry required to realize all dot operators. A Han Carlson configuration reduces the area overhead of a radix 2 Kogge Stone adder by removing every other connected node in the tree (resulting in a *sparseness* of two) as shown in Figure 3.8(c) [29]. The removal of all dot operators that generate odd-numbered carry out bits is compensated by relatively simple additional circuitry that ripples even-numbered carry out bits to their next stages (shown as dashed lines shaded in gray).

A second carry lookahead adder with a reduced area scheme is the Brent Kung adder, which computes only the carry out signals to bit positions of powers of two while realizing all others with an inverse binary tree [30, 27]. However, the resulting irregular layout of dot operators leads to complex wiring and varying gate fanouts, as illustrated in Figure 3.8(d). The inverse binary tree is shaded in gray and appears above the forward tree.

The radix 2 Kogge Stone is implemented in static, dynamic domino, and passgate styles while the Han Carlson, Brent Kung and radix 4 Kogge Stone architectures are implemented in static. Figure 3.9 shows circuit schematics for the dot operators, which create propagate and generate signals for these carry lookahead trees.

Overall, these eleven adders exhibit varying levels of circuit complexity, intermediate node capacitance, gate fanout, transistor stack heights, critical path lengths, and



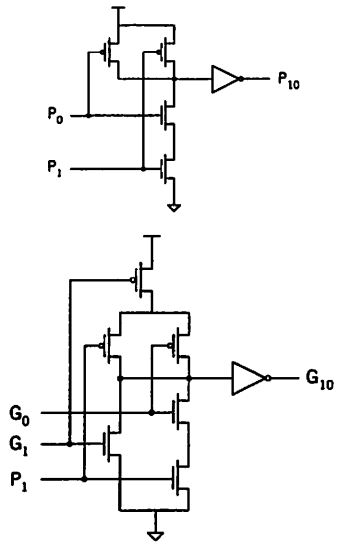
(a) Kogge Stone, radix 2

(b) Kogge Stone, radix 4

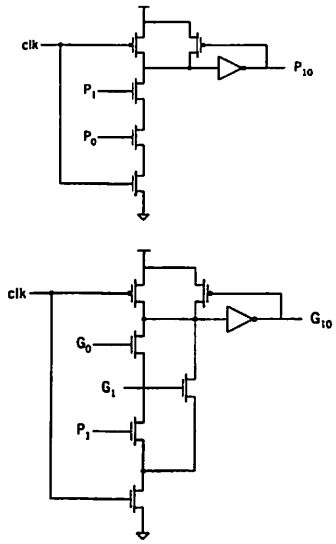
(c) Han Carlson, radix 2

(d) Brent Kung, radix 2

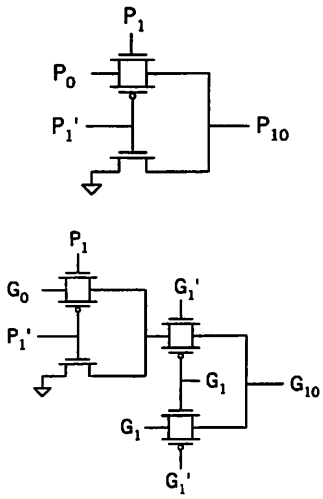
Figure 3.8: Various carry lookahead tree architectures for a 16-bit adder.



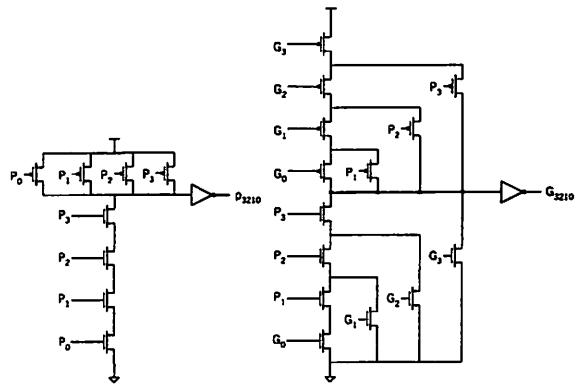
(a) Static radix 2



(b) Dynamic radix 2



(c) Passgate radix 2



(d) Static radix 4

Figure 3.9: Circuit implementations of dot operators used in carry lookahead trees.

intrinsic evaluation methods. Results from the Monte Carlo analysis form a comparison of the relative robustness of each to the effects of process parameter variations.

Optimization of Transistor Sizes

In order to conduct an unbiased comparison of the effects of process variability on designs within each circuit type, all transistor sizes are subjected to an objective delay optimization given a fixed set of area and timing constraints, and under similar loading conditions. The specifics of these constraints differ for the NAND chains and the adders, as dictated by differences in circuit complexity; however the goal of objective sizing remains consistent for both types. All circuits presented in this study are optimized using an in-house software routine implementing the Genetic Algorithm (GA), with the objective to minimize propagation delay while meeting specified area and timing constraints.

The Genetic Algorithm The Genetic Algorithm is an optimization routine inspired by the evolution of living species, in which a gene pool is defined by the collective strength of all its organisms, rather than by individual strengths and weaknesses. A global optimization technique may be likened to these biological theories: the global minimum of a set of values is not likely to be converged upon by pursuing progressively small individual numbers, because searches with such narrow scopes often become trapped in local minima. Instead, GA tracks entire sets of numbers that tend toward the minimum, and is thus an optimization methodology that produces a set of progressively optimal solutions rather than one that pinpoints a single best case.

The in-house software implementation of GA performs the optimization routine by externally processing the output data from large batches of SPICE simulations, whose

netlists are augmented with optimization-specific parameters and cost functions. A user-defined initial solution is exhaustively mated with successive generations of design values until the global minimum of the cost function is determined. The GA outputs are then input back to the original SPICE netlist, in the form of optimized transistor sizes.

Because the interface between GA and SPICE requires manual analysis and modifications, it is an aspect of the experimental setup that may benefit from a more seamless integration. Furthermore, the GA sizing algorithm does not consider the impact of fluctuations in process and operating parameters on delay spreads, producing optimal sizes for only the nominal case. The combination of the GA and Monte Carlo simulation environments into a unified, variation-aware sizing methodology may provide sizing guidelines that yield optimal performance for all variation scenarios.

Constraints imposed on all designs The first constraint imposed on each design is a gate area limitation. The area of each transistor gate is calculated as the product of its width (W) and length (L), and the total area is summed across all transistors in the design. This total gate area is constrained by a maximum area threshold, whose value is based upon parameters specific to the technology (e.g. minimum linewidths) and complexity of the design (NAND chain versus adder). The second constraint is to equalize the high-to-low and low-to-high signal switching delays of all designs, while minimizing the average propagation delay.

Based upon these objective conditions for area, timing and loading, all circuit designs are thus sized for optimal speed without benefiting from advantages due to area-delay tradeoffs or skews in timing edges. Therefore, all designs of a given circuit type are submitted to the Monte Carlo simulation framework as previously described,

and the resulting variability in performance metrics is thus compared across logic evaluation styles and circuit architectures without including any biases inherent to the particular design.

3.2 Results

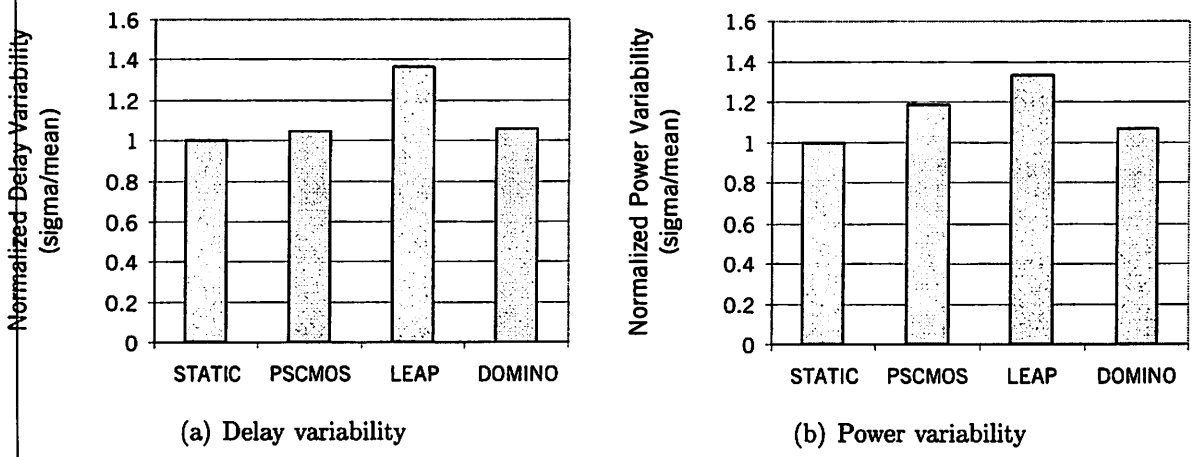
The results from all Monte Carlo simulations for both types of circuits (NAND chains and adders) are compiled and analyzed as follows. To compare the delay and power variabilities across the circuits when all parameters varied simultaneously (simulation type I), the standard deviation of the simulated delay and power spreads are first normalized to their corresponding raw mean values ($\frac{\sigma}{\mu}$). Then, in order to compare results across logic style for both circuit types without disclosing industry-sensitive data, these percentages are normalized a second time to the logic style displaying the least amount of variability for each performance metric.

A similar analysis is performed for results from the second type of simulation (II), in which parameters are varied individually. First, the total variability ($\frac{\sigma}{\mu}$) for each logic style is normalized to 100%, and then the individual percentage contributions of each of the five parameters (V_{th}, V_{dd}, t_{ox}, L and W) are determined.

3.2.1 Delay and Power Variability

NAND Chains

Figure 3.10(a) plots normalized delay variability for the four NAND chains loaded with static capacitive loads. The static CMOS implementation displays the most well-controlled delay variation levels, while the LEAP style suffers from 36% greater variability. The dynamic and pulsed static styles remain comparable to the static



(a) Delay variability

(b) Power variability

Figure 3.10: Normalized performance variabilities of NAND chain with static capacitive loading.

case with 6% and 5% higher delay variability, respectively.

Figure 3.10(b) plots normalized power variation levels for the NAND chains, and again illustrates the high robustness of the static CMOS implementation. In comparison, the LEAP style suffers the highest amount of relative power variability, at 34%, while the dynamic and pulsed static styles are 6% and 19% higher.

Adders

Simulation results for the family of 16-bit adders indicate that the static implementation of the carry-select adder is the most robust to delay variability, as shown in Figure 3.11(a). Furthermore, while variability levels for most other static and dynamic designs fall within 20% of the static carry-select, the passgate families clearly suffer from the least amount of variation control. The three designs with the highest relative delay variabilities are the static ripple carry adder with passgate-based Manchester carry chain (31%), the passgate implementation of the carry-select (50%), and passgate-based radix 2 Kogge Stone (67%), which is the highest of all. This re-

sult may be attributed to the worst-case assumption of perfect spatial correlation of V_{th} variation in all designs. The passgate logic evaluation style relies on a signal propagating along a series of pass transistors, with V_{th} variations combining at each stage along the passgate chain. In this study, an identical value is added at each stage, which likely sums to a large total variation magnitude. For a more realistic level of spatial correlation ($\rho_{V_{th}} < 1$), a random correlation component should be added to the V_{th} variation, allowing fluctuations to average along successive stages for a lower mean value. Further work is needed to understand true levels of spatial correlation thoroughly.

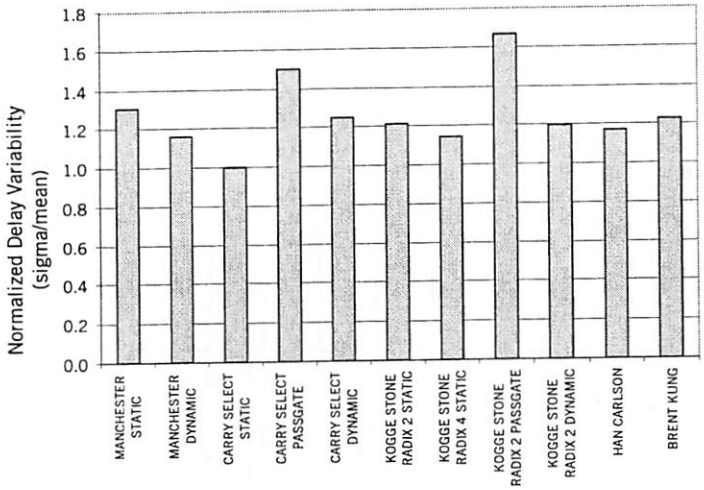
Trends in power variability for all adders is shown in Figure 3.11(b). The static Manchester carry chain adder displays the most predictable power values, while the relative variabilities of other designs range between 22% and 137% higher. The two designs least robust from a power perspective are the static, radix 2 Brent Kung and static, radix 4 Kogge Stone adders, each with over 100% larger spreads. This result may be attributed to the higher relative complexities of each of these designs, which both have large amounts of intermediate node capacitance along critical paths. The Brent Kung topology has widely varying internal fanouts at each node, characteristic of its irregular tree structure, with the largest capacitive load at its eighth bit position (Sum_{07} as shown in Figure 3.8(d)). Furthermore, the dot operator for the radix 4 Kogge Stone architecture has the highest transistor stack height of all designs (four each of PMOS and NMOS transistors, as shown in Figure 3.9(d)), resulting in a large capacitive load at each intermediate output node. These loads are composed of internal capacitances of active transistors, which fluctuate with variations in process parameters. During adder operation, the active power required to continuously charge and discharge these varying capacitances fluctuates correspondingly, resulting in the higher relative amount of power variability for these complex

architectures. In comparison, the more regular adder architectures display less spread performance ranges.

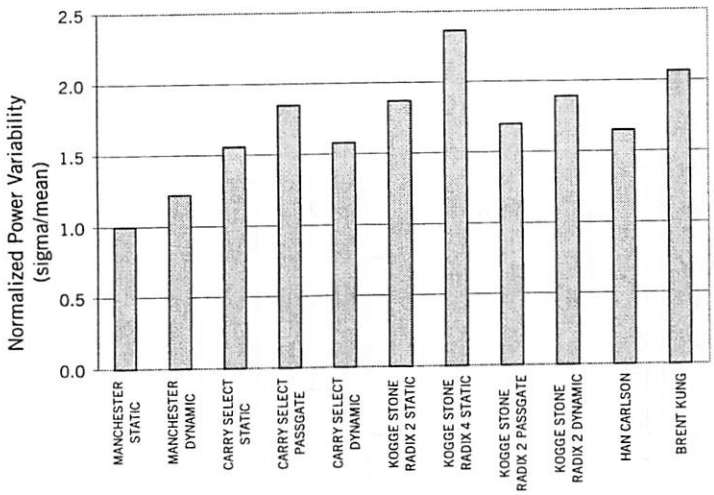
The power delay product (PDP) metric is a standard figure of merit that defines quality in a design by the amount of power required to achieve a given level of performance. The normalized PDPs of all adders are plotted in Figure 3.12. According to these results, the strongest designs fall within the group of carry lookahead architectures, with the passgate implementation of the radix 2 Kogge Stone adder exhibiting the smallest raw PDP value. This result is due to the superior performance of tree adder architectures, coupled with the increased speed of passgate logic evaluation.

When evaluating the adder with overall optimal performance, the variability in power and delay should be considered in conjunction with nominal PDP levels. For example, while the passgate radix 2 Kogge Stone implementation has the smallest nominal PDP, it suffers variability levels 30% higher in delay and 70% higher in power than the static ripple adder with Manchester carry chain. Figure 3.13 shows the normalized variability of these PDP values. Results showed that both the dynamic implementation of the carry-select and the static, radix 2 Han Carlson adders achieved low PDP variability, while almost all other designs fell within a range of 20% of these two. The only exception was found in the static Manchester ripple carry adder, with a PDP varying by almost 40%.

In general, the optimal adder design for a given application is chosen after considering variability levels simultaneously with raw nominal values. For example, if worst case values are the primary concern, an adder with a widely spread, low raw delay may be a better design than an adder with a more predictable but signicantly larger mean delay. While normalized delay and variation levels are not sufficient for assessing and identifying the optimal adder design, the raw values are proprietary to the industrial research site and are thus not shown.



(a) Delay variability



(b) Power variability

Figure 3.11: Normalized performance variabilities of 16-bit adders.

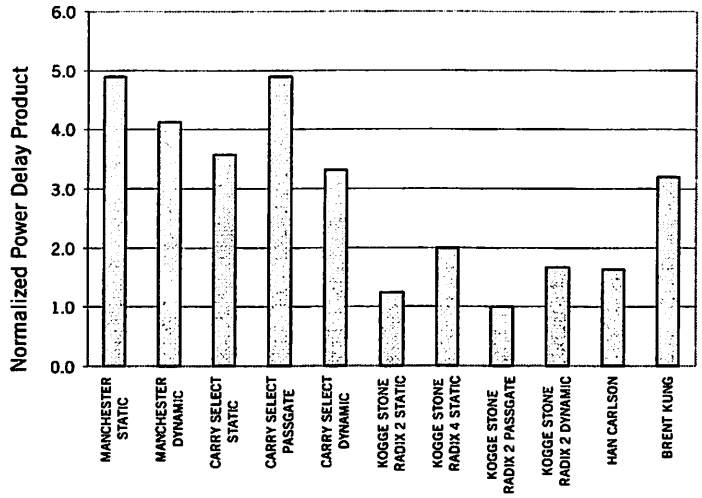


Figure 3.12: Normalized power delay product of 16-bit adders.

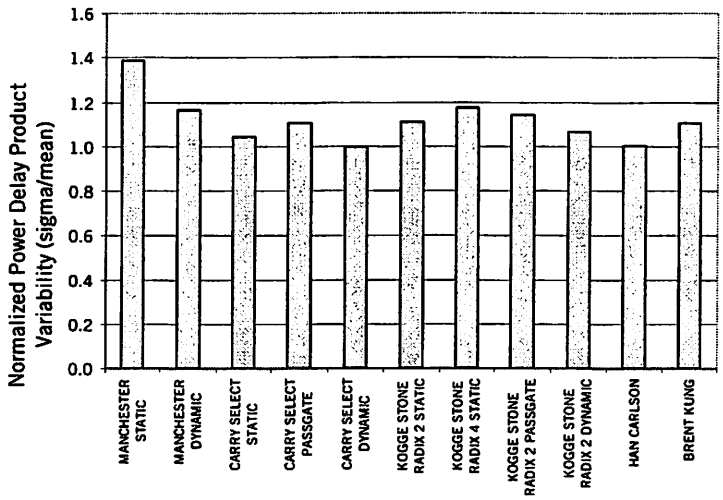


Figure 3.13: Normalized power delay product variability of 16-bit adders.

3.2.2 Individual Parameter Contributions

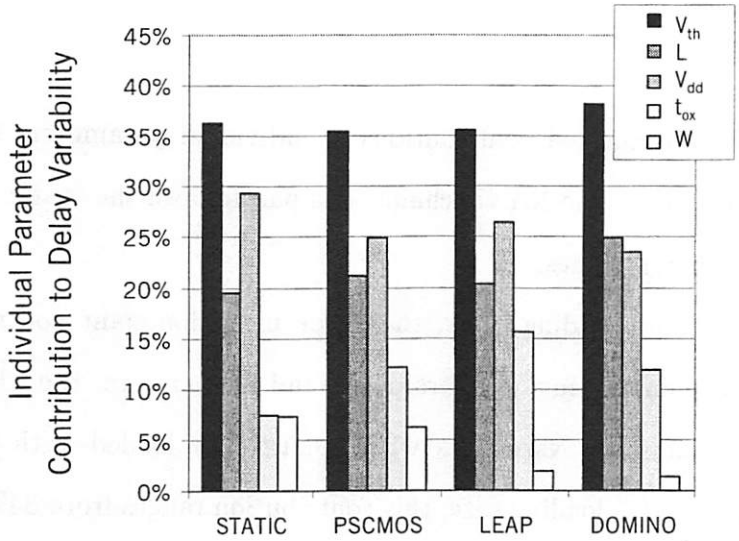
NAND Chains

Figure 3.14 plots the normalized contributions of individual parameters to overall levels of delay variability for the NAND chains, comparing both the static capactive loading and FO3 loading schemes.

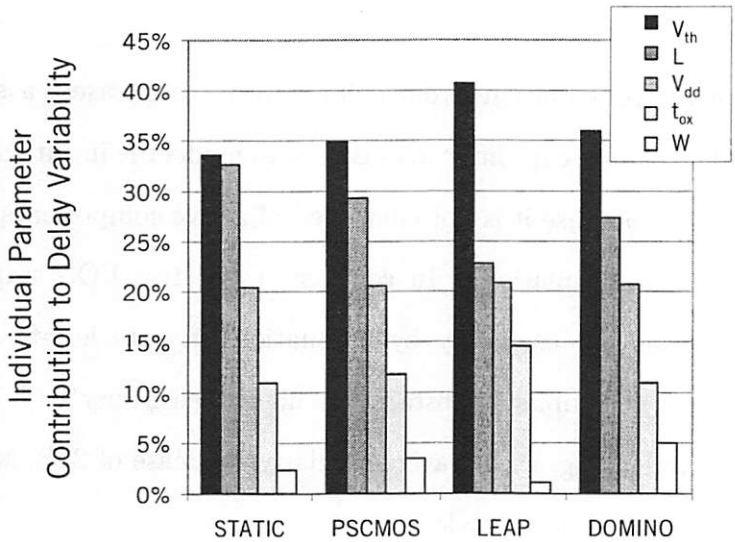
It is clear that in both loading cases, the single most dominant contributor to delay variability is the fluctuating V_{th} parameter, and furthermore, the LEAP style is most sensitive to these V_{th} variations when realistically loaded with an active successive stage. In the FO3 loading case, this contribution ranges from 34% to 41%, a slightly wider range than the 36% – 38% for static capacitive loads. This result is again likely due to the assumption of perfect spatial correlation for threshold voltages; identical V_{th} variations combine along each passgate stage to produce a worst-case total variability.

When comparing the contributions from L for both loading cases, a significant difference is noted. The passive capacitive load is used to model the input capacitance of a successive stage, but because it is not composed of active components, its value remains static through all simulations. In contrast, the active FO3 stage load is affected by process variations, especially by fluctuations in gate length variation. The FO3 loading case thus exhibits a substantially higher sensitivity to L variations than the fixed capactive loading, with a average relative increase of 22% and a peak relative increase of 40% in the static style.

Finally, the parameters least affecting delay variability are W and t_{ox} in both loading cases, with a total contribution of approximately 15% for both loading schemes. These results are consistent with the assumptions from Chapter 2 that gate length, threshold and supply voltages are the most significant.



(a) Static capacitive loading



(b) FO3 stage loading

Figure 3.14: Individual parameter contributions to delay variability of NAND chain.

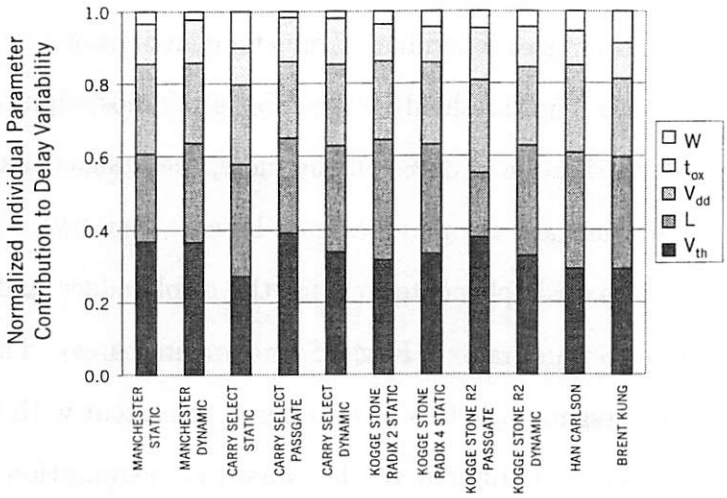
Adders

Figure 3.15(a) shows the normalized individual parameter contributions to delay variability for the 16-bit adders. The threshold voltage V_{th} is the most significant parameter, with an average contribution of 33%. Furthermore, the designs that are most sensitive to this V_{th} variation are the four passgate-based adder styles (the static and dynamic Manchester chain implementations for the ripple adder and the passgate styles for the carry-select and radix 2 Kogge Stone architectures). This trend of heightened sensitivity of passgate logic to V_{th} variation is consistent with the NAND chain results and may also be attributed to the worst-case assumption of perfect spatial correlation in V_{th} variation.

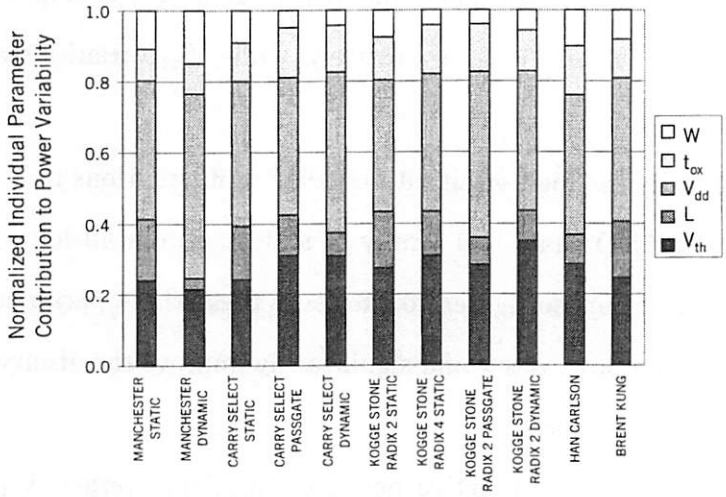
Effects of gate length L are nearly as significant as V_{th} contributions, accounting for a normalized 28% of the overall variability, due to variation-prone active transistor loads. Furthermore, process parameters t_{ox} and W are again the least significant, with average contributions of 5% and 10%, respectively, while V_{dd} variations account for the remaining 23%.

These results quantify the high sensitivity of delay to fluctuations in V_{th}, V_{dd} and L, consistent for the NAND chain and family of adders, across all logic evaluation styles. Clearly, efforts to impose tighter control over these three parameters during manufacturing and design processes would significantly improve the ability to control the range of transistor gate delays.

When considering variability in active power dissipation, average V_{dd} contributions dominate at 41%, as shown in Figure 3.15(b). Fluctuations in V_{th} are also significant, accounting for nearly 30% of the power spreads. Techniques for ensuring sufficient V_{th} control during manufacturing processes and for reducing V_{dd} noise during circuit operation are key for ensuring predictable power dissipation.



(a) Delay variability



(b) Power variability

Figure 3.15: Normalized performance variabilities of 16-bit adders.

3.2.3 Discussion

Simulated results showed that circuits designed in static CMOS generally display the highest levels of robustness to parameter variations, while passgate-based circuits suffer delay spreads from 30% to 70% higher than corresponding static implementations. Trends in power variability for the adder circuits suggest a dependence upon intermediate node fanout; designs with large amounts of fluctuating capacitance at internal nodes generally yield the least predictable power levels, while designs with both fewer transistors and more balanced internal signal fanouts display the least amount of power fluctuation.

Total variability levels are divided into the sum of five individual parameter contributions. The most significant contributors are identified as V_{dd}, V_{th} and L, accounting for an average of 85% of delay variability and, in the adder study, 80% of power variability. Among these three factors, V_{th} emerges as the most significant physical parameter affecting both delay and power, with contributions from L nearly as significant.

The total contribution of V_{th}, as well as its effect on passgate styles, may be overestimated in this study because perfect parameter correlation is assumed for all variable parameters. In reality, threshold voltages vary as a result of variations in channel doping concentrations, which are not correlated with physical proximity to adjacent transistors. These V_{th} distributions contain random components, which are likely to average along successive stages, rather than add identically. Further research is needed to more accurately model the spatial correlation between all parameters, in order to reconcile the true contributions from each parameter. This work provided a worst-case analysis, with a result suggesting that substantial benefits may be gained with improved manufacturing control of both V_{th} and L parameters.

3.3 Future Work

Previous investigations into achieving desirable performance levels in bulk CMOS designs through supply and threshold voltage optimization have been supplemented with studies on delay and power variability in pd-SOI designs. A family of representative circuits implemented in various logic topologies, including NAND chains and 16-bit adders, is first optimized for delay within an area constraint, then subjected to two sets of exhaustive Monte Carlo simulations.

While results from Chapter 2 and this work provided a foundation for studying the robustness of circuits to parameter fluctuations, future work would benefit from a number of refinements to the experimental setup and resulting analysis.

First, the performance metrics under study may be expanded beyond basic circuit delay and active energy and power dissipation. There are a number of other traditional circuit characteristics that define robust performance in low power operation, including immunity to noise and the amount of leakage power dissipated. In addition to these standard concerns, pd-SOI circuits are also uniquely affected by phenomena specific to its physical structure, due to the additional insulated layer buried in the silicon substrate. New effects emerge that require in-depth studies, including the transient "history effect" and dynamic threshold variation [23]. Currently the response of these metrics to variations in circuit parameters is not thoroughly understood and thus further research is needed to analyze this behavior.

Finally, optimizations in the simulation setup may potentially improve computational efficiency as well as produce more strategic timing margins. The current GA routine used to determine transistor sizing for optimal critical path delay considered only nominal parameter values, and was run independently of the Monte Carlo variation simulations. The parameter fluctuations were added only after these optimal

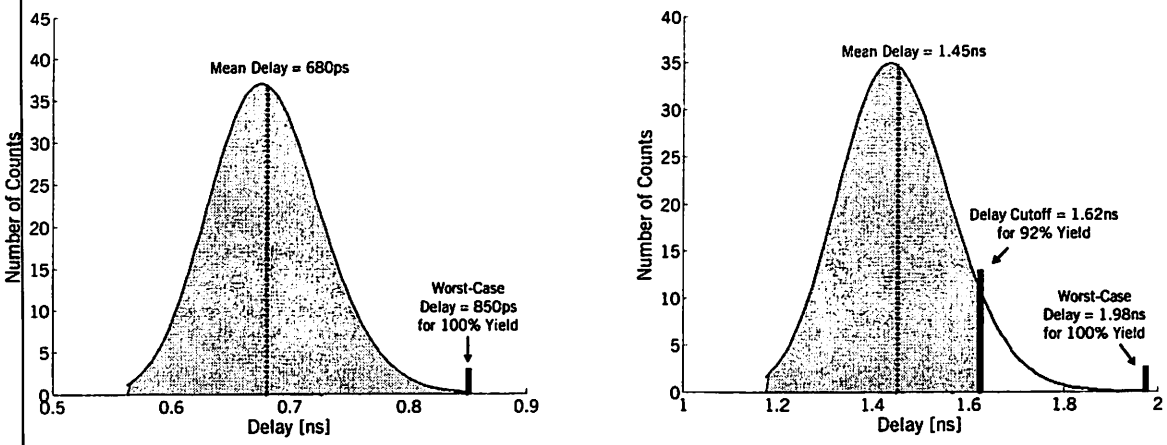
sizes are chosen, implying that variations in the worst case timing paths were not considered during sizing optimization. The combination of these two simulations into a unified, variation-aware sizing methodology may eliminate unnecessary design iterations and variation-induced timing faults by considering all timing corners that result from the full range of parameter fluctuations.

Chapter 4

Architecture Study: Robust Design of Finite State Machines

Chapters 2 and 3 motivated a new timing methodology for ultra low power designs; as circuits are scaled to increasingly smaller critical dimensions and are operated under aggressively reduced supply voltages, worst-case clocking methods may result in prohibitively slow operating frequencies due to drastically increased delay variability. Results from Monte Carlo simulations of a static 4-bit adder from Chapter 2 demonstrate that there is a significant tradeoff between performance and energy dissipation; an 8% relaxation in circuit-level yield may lead to 92% savings in active energy, while maintaining adequate performance levels for ultra low power applications.

To illustrate this point, Figure 4.1 plots fitted lognormal distributions to delay data for the static 4-bit adder operating under nominal and reduced voltage conditions. Under nominal V_{dd} and V_{th}, Figure 4.1(a) shows that 100% yield is achieved by imposing the worst-case delay cutoff of 850ps, with an active energy dissipation of 312fJ. Although the technique of aggressively scaling the supply and threshold voltages is effective for reducing energy dissipation, it causes a penalty of increased delay magnitudes as well as a wider spread of their values. Figure 4.1(b) shows that although the active energy dissipation has decreased to 54fJ, the lognormal shape of



(a) Nominal voltages: V_{dd} = 1.2V, V_{th} = 240mV (b) Reduced voltages: V_{dd} = 300mV, V_{th} = 40mV

Figure 4.1: Fitted lognormal distributions to delay data for static adder at nominal and reduced voltages.

the distribution has a long trailing tail, resulting in a 132% increase in the worst-case value. Because there are actually relatively few points under the length of this tail, the cutoff may be set at a smaller delay without a significant sacrifice in yield. In this example, only 8% of all simulated delays fall between the range of 1.62ns – 1.98ns; if the cutoff is instead chosen at 1.62ns, the resulting yield is still a relatively high 92%, and the operating frequency increases by 22%.

In general, as technologies continue to scale, variations in physical process parameters and operating voltage values will become increasingly worse, causing gate delays and delay spreads to increase dramatically. This increased variability will result in delay distributions with prohibitively long trailing tails, more exaggerated than in this example, motivating the need to abandon worst-case timing techniques for a more efficient tradeoff between clock frequency and yield.

The objective of this work is to intentionally design for small losses in yield, in the interest of maintaining adequate performance in ultra low power design. These

sacrifices manifest as timing errors at the transistor level, due to clocking the design faster than the worst case. It should be emphasized that the nature of these errors is static; once the circuit is manufactured, each transistor comprises physical device parameters including gate length, intrinsic threshold voltage, and oxide thickness, which are set during the manufacturing process. This implies that repeated performance measurements of the transistor under the same temperature and bias conditions produce the same values each time, neglecting device breakdown. Although it is possible for dynamic errors to also affect these circuits, the likelihood is not increased as a result of the timing methodology, and thus only static errors are studied in this work.

The overall goal of this research is to explore a high-level fault tolerant methodology to compensate for timing errors that occur at the transistor level. The representative system under study is a finite state machine (FSM) controller, in which timing errors are modeled as unreliable state transitions and faulty outputs. Of all blocks that compose a complete system, the FSM controller is chosen for study due to its critical role in dictating all aspects of proper functionality; its task of managing the interactions and flow of operations between all subblocks is of utmost importance. Because faulty controller behavior is likely to cause fatal errors in an entire system, all errors that may occur must be both detected and compensated.

In this work, a fault tolerant scheme is proposed for detecting and repairing erroneous state machine behavior, and a comparison between two error compensation methods is investigated. The next section offers a brief background in related research, followed by the introduction of the proposed methodology in Section 4.2. Section 4.3 details all components of the experimental setup, including the FSM under study and the logic synthesis tool. Results and analyses are presented in Section 4.4, concluding with directions for future work in Section 4.5.

4.1 Previous Research

Previous efforts to design systems that are immune to static errors have resulted in fault tolerant design techniques at transistor, circuit and architecture levels.

At the lowest level, a solution proposed for improving the performance of a circuit in the post-fabrication stage is known as Adaptive Body Biasing [31], in which the propagation delay of a gate is controlled via the body terminal of the transistor. Forward body bias voltages are applied to the slow transistors along critical paths such that their speed is increased sufficiently to meet timing constraints. However, the range of possible performance improvements is limited by physical device concerns (e.g. the amount of forward biasing must not exceed the turn-on voltage of substrate pn-junction diodes) and the need to discretize the range of available body bias voltages in order to practically implement this technique. Furthermore, this approach is at a low level in the design abstraction hierarchy, requiring an exhaustive search of slow transistors along critical paths. For VLSI designs comprising tens to hundreds of millions of transistors, the granularity of this technique is prohibitively fine.

At a higher level, a number of error control mechanisms for datapath circuits have been proposed, such as a digital filter IC designed with an Algorithmic Noise Tolerance (ANT) scheme [32]. The nature of the ANT method is heavily dependent on the predictive nature of the filter outputs; the filter tap coefficients describe a correlation between successive output values, and the linear forward error detection simply performs a pair-wise comparison on output values to detect inconsistent patterns. Another typical datapath circuit with regularities in its output behavior is the unsigned adder: the value of the most significant bit (MSB) likely remains constant for consecutive clock cycles and thus a simple fault correction scheme may detect random toggling in the MSB as erroneous behavior.

In comparison with datapath elements, a typical FSM controller is often simultaneously communicating with a number of distinct blocks in the system and its behavior may vary widely depending on the application. Moreover, when logic evaluation errors strike controller elements, they likely result in both undesired state transitions and erroneous output values, manifestations that are random and unpredictable. Therefore, methods of designing fault tolerant schemes for FSMs that do not rely on regular output patterns must be considered.

A classic system-level approach to fault tolerant FSM design is the Triple Modular Redundancy (TMR) scheme, in which three identical copies of an FSM run simultaneously with a majority vote arbitrating the correct output value. This mechanism works well for dynamic errors in critical systems because the likelihood of multiple dynamic hazards occurring in a small vicinity is extremely small (e.g. external particle radiation that may strike space-borne electronics [33]). However, the TMR technique is not applicable to repairing static errors in FSMs because an error occurring in one copy is equally likely to manifest in the other two.

The following proposed fault tolerant FSM methodology is motivated by the relatively sparse body of research of high-level approaches for compensating static timing errors.

4.2 Proposed Solution

The principle behind the proposed solution is the classic control theory of Engineering Change: a module known to be faulty, but whose intrinsic structure cannot be altered, may be controlled by an external block such that their combined behavior meets a designated specification [34].

In this work, a benchmark FSM control system is selected in which energy dissi-

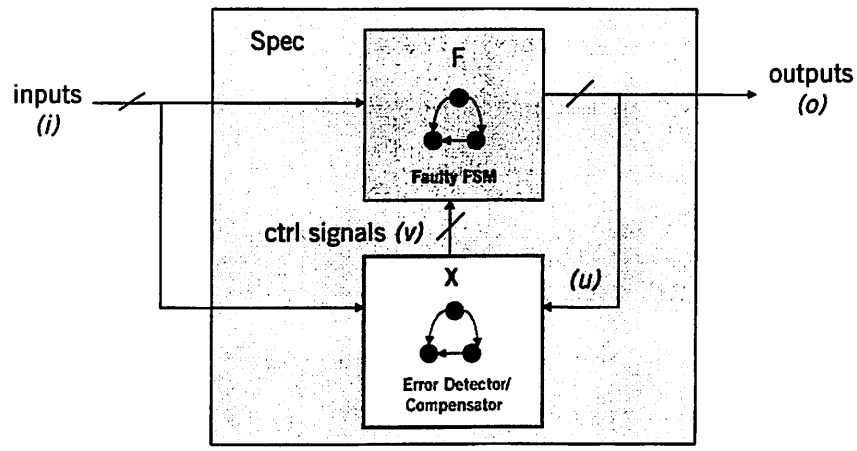


Figure 4.2: Topology of proposed error compensation scheme.

pation is reduced by significantly lowering the power supply voltage from which all circuits operate. As previously discussed, the resulting increase in delay variability is likely to cause faulty system behavior. An external piece of hardware is thus added to control the faulty FSM and compensate for undesired state transitions and outputs. The desired, specified behavior is exactly that of the original FSM operating under nominal supply voltage, which produces the correct sequence of output signals for a given input pattern. Figure 4.2 illustrates the proposed solution topology: the faulty FSM F is controlled by an external error compensator X, such that the correct outputs o are produced for each set of inputs i, according to a specification *Spec*. Furthermore, F communicates with X through a set of signals u, and X sends appropriate control signals v to F. The nature of this approach dictates that the module X be absolutely free of errors, while the control signals that are added to the original FSM may actually experience or even cause errors, because they will ultimately be a part of the faulty network.

An application of this proposed error control method is for repairing the behavior of a circuit, which may or may not be faulty, after it is manufactured. If a small yield

loss (i.e. due to static timing errors) is detected for the circuit, the error compensator circuit may be added to its operating environment, to add external control for correcting erroneous behavior. The target implementation of the error control block is such that it has a minimal number of pinouts and communicates only with the faulty block; corrected ouput signals are still generated and produced by the original block so that minimal disruption is caused to the embedded application environment. Ideally, because the error control block must exhibit error-free behavior, it would operate under nominal V_{dd} and have minimal area and energy overhead. Specifically, the total energy consumed by the composition of the error compensator block and the FSM operating under low supply should be less than the energy consumed by the original FSM operating at nominal supply. Furthermore, in order to repair the maximum number of possible errors that may occur, the behavior of this compensator circuit may be programmable so that its functionality conforms to faults specific to each manufactured circuit.

4.3 Experimental Setup

The experimental setup may be described in four parts: the finite state machine under study, the modeling tool, the error correction schemes and the error injection method.

4.3.1 FSM Under Study

The FSM controller studied in this work is derived from a digital circuit used in a network protocol for PicoRadio, an ultra low power, wireless sensor network designed by researchers at the Berkeley Wireless Research Center [35]. Specifically, the controller is one of five that compose the locationing subblock, which performs a least sum of

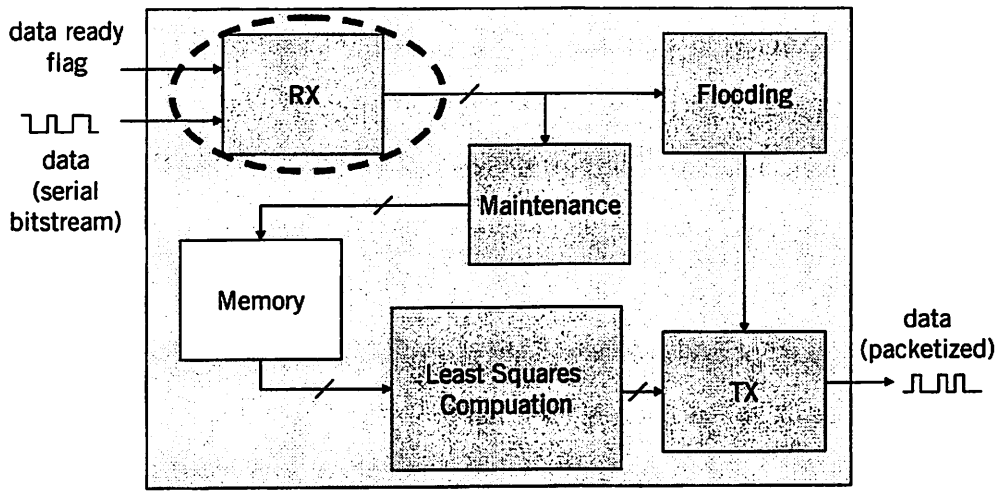


Figure 4.3: Block diagram of embedded locationing engine within the PicoRadio charm chip (FSM under study circled).

squares computation to resolve the x, y and z coordinates of each sensor node. Figure 4.3 shows a block diagram of this locationing engine, designed and implemented by Tufan Karalar. The representative controller under study in this work is the receive (RX) subblock (circled in the diagram) and its state transition diagram (STG) is shown in Figure 4.4. This block performs a standard de-packetization algorithm on a serial bitstream of data; when data are ready to be received, the FSM exits the default idle state, processes the data by visiting the other four states in succession, and then returns to the idle state to wait for the next set of data. This return to idle condition is strictly enforced, such that the FSM is not considered to function properly if ever deadlocked in any state.

Although this controller is relatively small when compared to those in larger scale systems, its behavior is representative of FSMs in more complex designs and its simplicity lends itself well for use as a case study. Furthermore, its functionality is of critical importance to the sensor network as a whole; it serves as the interface between the physical layer and the remainder of the locationing block. The successful

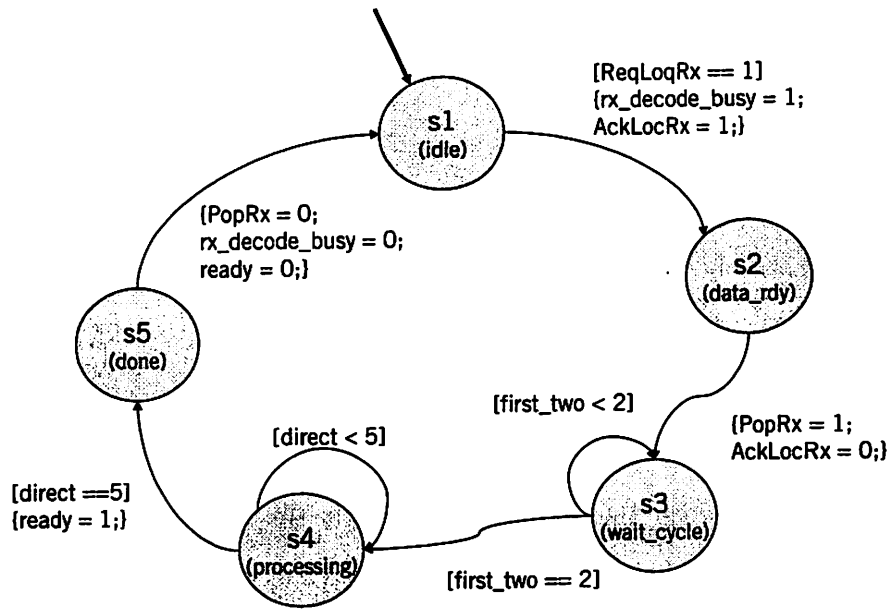


Figure 4.4: State transition diagram of RX subblock.

flow of network data traffic is entirely dependent upon a functioning receive block to interpret and forward all data packets to their proper destinations.

4.3.2 Modeling Tool: MVSIS

A language equation may be used to formally define the relationship between the faulty FSM F, the error control module X, and the specification $Spec$ that their composed behavior should meet:

$$F \circ X \subseteq Spec \tag{4.1}$$

where

- F represents the unalterable circuit implementation for the faulty FSM, operating under reduced V_{dd}, augmented with external control signals v as inputs
- $\circ$ represents "the composition of"
- X represents the error compensator module

- $\subseteq$ represents "conforms to"
- *Spec* represents the proper input/output specification of the FSM under nominal V_{dd}

MVSIS is a multi-valued logic synthesis tool with language equation solving capabilities [36]; given the behavior of F and *Spec* as inputs, it synthesizes the most general solution (MGS) of the behavior of X. If the MGS exists, it encompasses the set of all possible behaviors of X in order to function as an error control module, thus indicating the success of the attempted control scheme. On the other hand, an empty solution space for X signifies irreparable faults in the FSM.

There are numerous ways to represent a set of logic functions in MVSIS; the two most commonly used are the behavioral and structural representations. Both representations require the logic function to be described in Berkeley Logic Interchange Format (BLIF), a text-based representation of an arbitrarily complex logic-level hierarchical circuit [37]. Details of these two types are now described.

Behavioral Representation of RX

The state transition graph of the RX controller from Figure 4.4 may be translated into a BLIF file containing truth tables that describe the state transition relations and output values. The result is a behavioral representation of the FSM, as illustrated in Figure 4.5. The state bits are shown in order of least to most significant from left to right.

Structural Representation of RX

The FSM controller may also be represented and manipulated in BLIF as a structural netlist of arbitrarily complex gates and latches, functioning as a canonical form for a given logic function. The structural representation is a network of binary 2-input

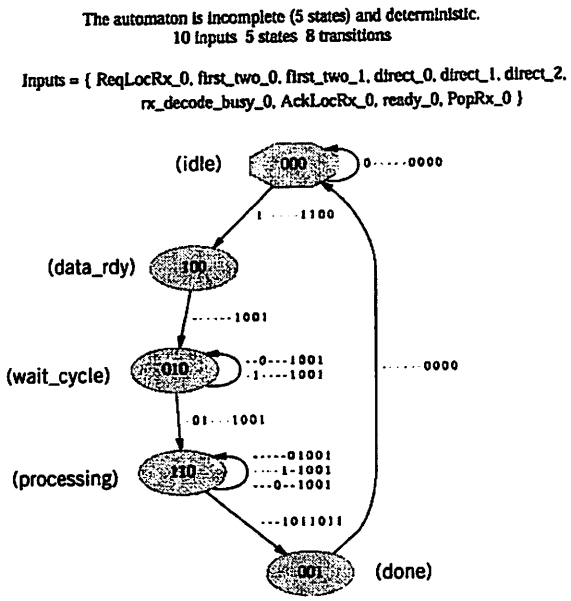


Figure 4.5: Behavioral representation of RX controller.

AND gates, with inverters placed between nodes as necessary, although not shown in the illustrated representation for simplicity. This network mapping is performed using the synthesis algorithm described in Table 4.1, with corresponding MVSIS commands listed next to each step. In general, this algorithm may be used to generate an optimized structural representation for any multi-valued FSM.

The optimized structural network of the RX controller consists of 18 nodes, 20 2-input AND gates and 3 latches, and is depicted in Figure 4.6.

Table 4.1: MVSIS synthesis algorithm to produce an optimized structural representation from a behavioral specification.

Synthesis Step	MVSIS Command
o encode latches into binary gates	`encode`
o encode I/O into binary gates	`io_encode -l`
o determinize the resulting binary network	`dize`
o optimize the nodes	`mvsis.rugged`
o perform a mapping into 2-input AND gates	`strash`

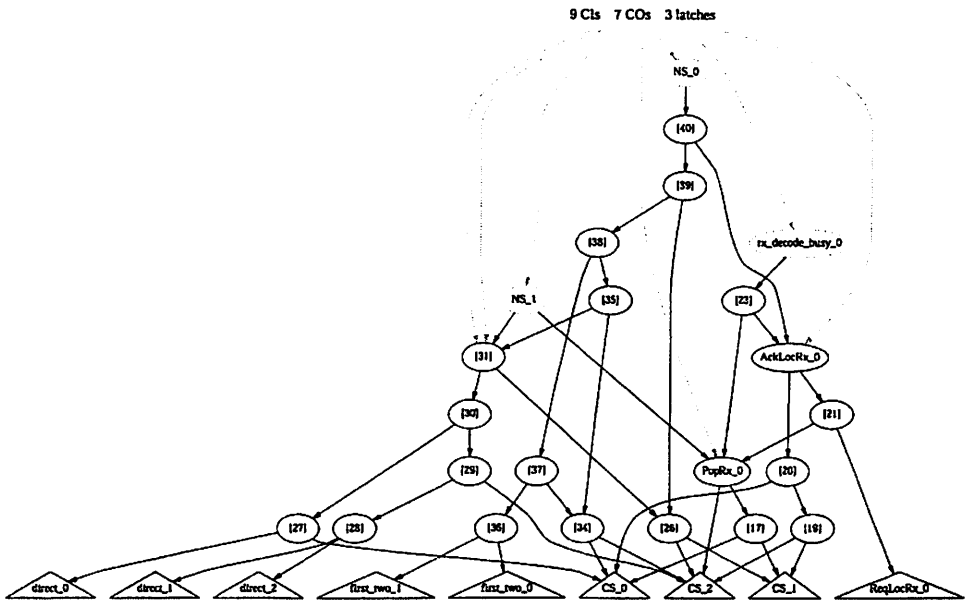
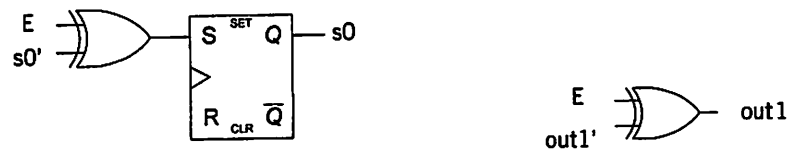


Figure 4.6: Structural representation of RX controller.

In order to compare multiple approaches for adding fault tolerance into the FSM, control signals are added to the original RX controller at both the behavioral and structural levels, and the effectiveness of each approach evaluated. These error control schemes are designed to repair undesired state transitions as well as faulty output values; details of their implementations are now discussed.

4.3.3 Error Correction Scheme

The RX FSM may be described as a Mealy machine, in which output values are functions of the current state as well as the input. Because output signals may change independently of state transitions, these two types of errors are treated separately. In all cases, the error control mechanism controlling the behavior of the RX FSM is dependent upon one or more external enable signals, whose values are input from and set by the error compensation module X. Techniques for structural level control



(a) For state transitions (b) For output values

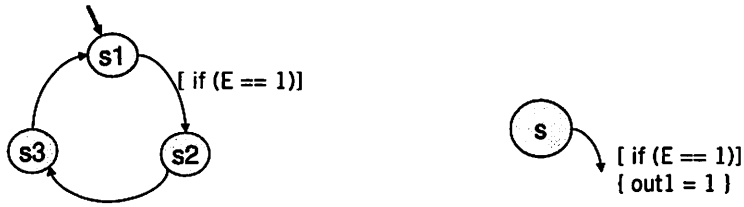
Figure 4.7: Methods for adding error control at the structural level.

of state transitions and outputs are now discussed, followed by methods for control at the behavioral level.

Structural Level Error Control

The current state of the FSM is held by three latches at the structural level, one for each state bit. In order to add control and correct for wayward state transitions, a 2-input exclusive-or (XOR) gate is inserted between the next state bit latch and the input to the latch, as shown in Figure 4.7(a). The second input to the XOR is an enable signal E: if the value of E is 0, the XOR functions as a buffer, passing the next state bit value through to the latch. Alternatively, if the value of E is 1, the XOR toggles the value of the next state bit. All seven combinations of adding error control to state bit latches are attempted: three cases with one enabled latch at a time, three with two latches at a time, and finally, the case of adding error control to all three latches. In all cases, a unique enable signal is added for each latch.

Figure 4.7(b) shows that the outputs are controlled in a similar manner at the structural level; a 2-input XOR is added to the output node with a signal E enabling or disabling the inverter functionality. For all attempted error schemes for output variables, only one output signal is enabled at a time.



(a) For state transitions (b) For output values

Figure 4.8: Methods for adding error control at the behavioral level.

Behavioral Level Error Control

The control scheme for state transitions at the behavioral level is implemented by enabling individual arcs in the STG, as shown in Figure 4.8(a). Figure 4.8(b) illustrates the mechanism for repairing output values; the enable signal E forces an output to be set to 1 regardless of the accompanying state transition.

4.3.4 Error Injection Scheme

The simulation and evaluation of these error correction methods is illustrated in Figure 4.9; structural level error control is shown on the left and behavioral level on the right. A naming convention is used to distinguish the two types of BLIF files: all behavioral representations (state transition graphs) are given the extension `.aut` while `.blif` is assigned to all structural representations (netlist of gates and latches). In order to model the timing errors in the FSM at a level as close to the physical device representation as possible, the simulated errors are injected at the structural level (`.blif`) in both cases.

For attempted repairs at the structural level, the gate netlist representation of the RX controller (`rx_orig_s.blif`) is the starting point for adding error control. First, the netlist is augmented with one or more XOR gates, each of which functions as a

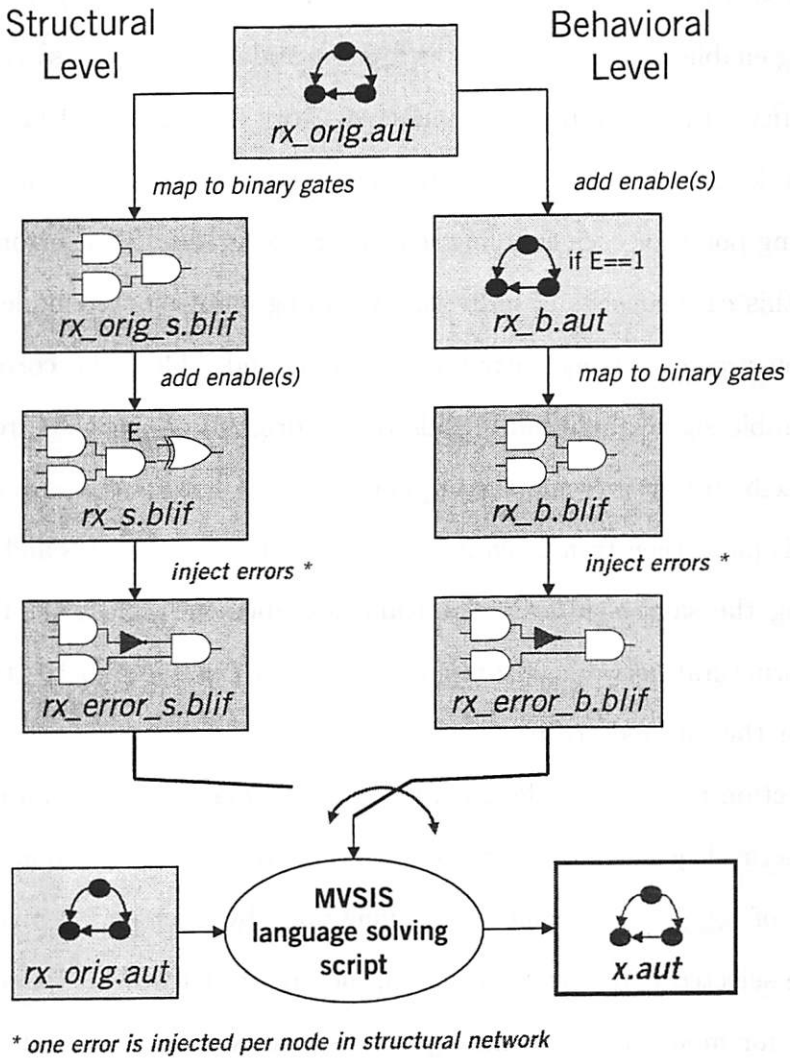


Figure 4.9: MVSIS-based simulation flow describing the method of comparing behavioral and structural level error compensation schemes.

programmable inverter that toggles output or state bits when enabled. The number of added XOR gates, and corresponding unique enable signals, varies between 1 and 3 in a total of eleven experiments; the effectiveness of each approach is compared with the amount of overhead incurred by adding these extra control gates. After the corresponding enable signals are added as input signals (forming `rx_s.blif`), the errors are then injected into the resulting binary network (`rx_error_s.blif`).

For behavioral level repairs, the truth table representation of the RX controller is used as the starting point for each enabling scheme (`rx_orig.aut`). The error control methodology in this case consists of individually adding one or more enable signals to state transition arcs or setting output values to 1, and adding the corresponding number of enable signals as input signals to the original design. The resulting behavioral file (`rx_b.aut`) represents the original RX FSM with the external enable signals added; this file is then translated into a structural netlist of gates and latches (`rx_b.blif`) using the same synthesis algorithm described in Table 4.1. This optimized binary structural network of 2-input AND gates (`rx_error_b.blif`) is the insertion point for the injected errors.

The error injection technique is designed to be a systematic sweep of all internal nodes of the structural network; one erroneous file is created for each node. Each error file consists of toggling the value of a unique node by inserting an inverter at the output of the selected node, representing an incorrectly latched bit value. Note that this method for modeling static timing errors is fairly simplified, as it assigns exactly one incorrect bit value at a time in the circuit. In reality, the percentage of transistors in the circuit failing to meet the timing specification and furthermore latching an erroneous bit is small; average yield losses are expected to be only 8% – 9%, as previously discussed. Furthermore, the mapping between transistor-level timing errors and their manifestations in state machine behavior is much more complex

and unpredictable than that suggested by this model. In reality, the structure and configuration of physical circuits composing a controller system depend on a number of factors: the state bit encoding method, the level of boolean logic optimization, and the place-and-route algorithm used to realize the logic function into gates. It is not possible to realistically model these complicated interactions between layers of design and thus a simplified error model is used.

Each of the erroneous structural netlists `rx_error_b.blif` and `rx_error_s.blif` represents a version of the unalterable, faulty FSM F. Once input to the MVSIS language equation solving script, it is extracted into a behavioral level, STG representation. The second input to MVSIS in all cases is the behavioral representation of the original RX controller operating under nominal V_{dd}, with no enable signals added and no errors injected, dictating the specification *Spec*. Given these two inputs, the language equation solver attempts to produce a set of behavioral solutions X, which contain all possible realizations of the error control module. If a non-empty X is produced, it is considered a successful fix of the unique error that appears in the faulty FSM F. Each particular solution $\hat{X}$ in the set X represents a controller for F, such that their compositional behavior $F \circ \hat{X}$ meets the specification *Spec* under all input and output conditions. For simplicity of calculating the cost of implementing the error compensator, the particular solution $\hat{X}$ is chosen from each MGS X by simply discarding the don't-care (DC) state without attempting to optimize the solution. Although this technique provides a simple estimate of the particular solution and its STG, it is clearly not an ideal method of extracting the most cost-effective solution; further work in the area of optimizing particular solutions from a MGS is needed.

The cost-effectiveness of each error correction scheme is measured by weighing the number of errors successfully repaired with the average overhead incurred for its implementation. Each node in the structural gate netlist of a design represents a

2-input AND gate and thus the total number of nodes is a metric for an approximate area comparison. The overhead is calculated by comparing the netlists of original, standalone design (`rx_orig.blif`) and the netlists composing the fault tolerant design. The total number of nodes in the fault tolerant design consists of the nodes in the error-compensated F netlist added to the average number of nodes in the netlists of all particular solutions of the error compensator $\hat{X}$.

4.4 Results

A comparison of repairing faulty output values and undesired state transitions at both the behavioral and structural levels is now performed.

4.4.1 Repairing Faulty Output Values

Table 4.2 compares the effectiveness of error control techniques at the structural and behavioral levels, for which methods of repairing all four output values are investigated. In three out of the four cases, the structural level repair of simply adding a single 2-input XOR gate with one externally controlled enable signal is both more effective and less costly than adding the ability to set the output bit to 1 at the behavioral level. The effectiveness of structural level error compensation methods ranges between 28% – 39%, with overhead penalties ranging from 235% to 260%. The only reasonably successful repair scheme at the behavioral level fixes only 20% of errors at a 230% penalty.

It is clear that for repairing errors in output signals, fault compensation is most effectively added to structural level netlists of gates and latches.

Table 4.2: Results for error correction methods for faulty output values at both structural and behavioral levels. All schemes used one enable signal only.

Output Variable	Repair Method	RX w/ Enable Avg # ANDs	$\hat{X}$ Avg # ANDs	% Repaired	% Overhead
PopRx	*struc*	*57*	*26*	*39*	*260*
	behav	—	—	—	—
Ready	*struc*	—	—	—	—
	behav	53	22	20	230
AckLocRx	*struc*	*51*	*26*	*39*	*235*
	behav	54	22	5	230
rx_decode_busy	*struc*	*53*	*24*	*28*	*235*
	behav	—	—	—	—

Table 4.3: Results for attempted repairs of faulty state transitions at the structural level.

State Bits w/ Enable Signals	# Enables	RX w/ Enable Avg # ANDs	$\hat{X}$ Avg # ANDs	% Repaired	% Overhead
NS_0	1	23	42	39	180
NS_1	1	—	—	—	—
NS_2	1	—	—	—	—
NS_0, NS_1	2	26	58	44	265
NS_1, NS_2	2	—	—	—	—
NS_0, NS_2	2	28	42	39	200
NS_0, NS_1, NS_2	3	31	48	44	245

4.4.2 Repairing Undesired State Transitions

Table 4.3 summarizes the effectiveness of using error correction techniques to control state transitions at the structural level; Table 4.4 summarizes the results at the behavioral level.

The most cost-effective scheme for repairing state transitions at the structural level consists of adding one enable signal to toggle the least significant state bit NS_0. This error correction scheme is 40% successful at a cost of 180%, the lowest penalty incurred in all cases. In comparison, the method of individually enabling the

Table 4.4: Results for attempted repairs of faulty state transitions at the behavioral level.

Transitions Enabled	# Enables	RX w/ Enable Avg # ANDs	$\hat{X}$ Avg # ANDs	% Repaired	% Overhead
s1→s2	1	29	56	23	270
s2→s3	1	26	51	22	240
s3→s4	1	27	54	28	255
s4→s5	1	31	47	19	240
s5→s1	1	27	54	28	255
CS→NS	1	31	61	27	300
CS→NS, rx_decode_busy	2	33	59	31	300

other two state bits NS_1 and NS_2 yield zero success rates. This lack of success may be understood when considering that toggling the two higher order bits results in transitions to states that would never be reached in the original RX controller, namely state 7 (011) and state 8 (111). Therefore, the technique of allowing the STG to reach these states does not benefit its fault tolerant behavior, while the other state transitions enabled are not necessary for compensating errors that occurred. On the other hand, due the sequential nature of all state transitions in the FSM under study, the addition of one enable signal for toggling the least significant state bit NS_0 proves to be an effective method for guiding faulty transitions back to their intended destinations.

The remainder of results at the structural level are also quite promising. Adding two independent enable signals to toggle both NS_0 and NS_1 as necessary is 44% successful at an average cost of 270%, whereas adding a third independent enable signal to control NS_2 is equally effective at a lower average cost of 245%. This implies that although the latter case requires more overhead for enabling a third state bit, its set of error compensator solutions $\hat{X}$ is on average less complex than those for two enabled state bits.

Results show behavior level control of state transitions to be significantly less effective than at the structural level. Schemes of enabling one state transition at a time results in success rates ranging from 19% – 28%, at costs of 240% – 270%, which are not competitive with results from enabling individual state bits. The scheme of allowing any state to transition to its next state (CS→NS) with a second independent enable signal controlling the value of the output variable (rx_decode_busy) is the most effective behavioral level scheme (31%) at a high cost of 300% overhead. Removing the ability to control the output and only allowing any state to transition to its next state is marginally less effective (27%) at approximately the same average cost.

4.5 Future Work

Simulations run with the aid of MVSIS to explore fault tolerant FSM design techniques show error control capabilities built into the structural level to be significantly more effective and less costly than similar efforts at the behavioral level. Specifically, the most cost-effective approach for controlling undesired state machine behavior is to add a single XOR gate and an accompanying enable signal into the netlist of gates and latches in order to toggle the least significant state bit when necessary. This specific control method is the most effective because of the sequential nature of the state transistions in this case study; for control systems comprising random state transitions, an approach in which multiple state bits are simultaneously enabled is expected to be more effective.

The types of error correction schemes investigated in this study are not expected to repair 100% of all possible timing errors due to yield loss and voltage overscaling. The study is intended to provide a comparison between two different sources of error control with results used as a guideline for further research efforts. In future studies

of the same nature, it is hoped that the success rate of similar fault compensation schemes may be increased to render them feasible in actual designs. If a large enough percentage of circuits may be designed tolerant to static timing faults and programmable error correction modules may be synthesized without incurring a prohibitively large overhead, the energy savings gained by voltage overscaling may outweigh the loss of discarding the small percentage of fatally crippled circuits.

There are many directions for improvement to the experimental setup described in this work, and more promising results are expected with optimized simulation steps. For instance, the extraction of a particular solution $\hat{X}$ from the most general solution X was not optimized, and may have contained a number of logic redundancies that contributed unnecessary overhead. Furthermore, static timing errors originating at the transistor level were simulated in the FSM behavior by toggling signal values at nodes in the gate netlists; this error injection algorithm was simplistic and may not have accurately modeled the manifestations of low-level error sources. In addition, the benchmark FSM under study consisted of only five states, containing successive state transitions and self loops. Its simplistic structure may not have benefitted from the attempted error control schemes; alternatively the set of possible errors generated may not have proven to be enough of a challenge for the error control schemes.

Chapter 5

Conclusion

In a perfect manufacturing process, all transistors would be fabricated with infinite accuracy according to design specifications; in the real world, robust design is the engineering method for coping with the variations in non-ideal processes.

This work investigated robust design techniques for ultra low power systems, introducing parameter variations at process and circuit levels and investigating methods for managing performance variability at all levels. Further research efforts should focus on determining the most suitable method of robust design at each layer of the design hierarchy, such that the synergy of all solutions produces an optimized, cost-effective robust design methodology that addresses all variation sources. Moreover, the most effective method for creating this methodology is to build successively upon low level models until a high level understanding that encompasses all phenomena is created.

At the lowest level, as discussed in Chapter 2, an understanding of spatial correlation between transistors is crucial for relating parameter matching between transistors as a function of distance. A model for spatial correlation for all parameters may be included in circuit level analyses, which in turn may help predict the probability of timing errors occurring at the logic level. This error probability may then be interpreted and statistically modeled at architecture and algorithm levels, so that the

variations that originally occurred due to manufacturing inaccuracies may be ultimately understood and corrected by system-level design techniques. The ability to bridge the gaps between abstraction layers and model the manifestations of variation at each layer is key for a truly robust design.

As forecasts on the fundamental limit to pure device size scaling loom in the foreseeable future, device and process innovation become increasingly more important for sustaining current scaling trends. Regardless of the implementing technology, the success of future nanometer designs will rely on performance models that accurately predict circuit delay, energy and power, in addition to novel robust design techniques to guarantee proper functionality across all variation corners.

Bibliography

[1] R. W. Keyes, "Physical limits in digital electronics," *IEEE Journal of Solid-State Circuits*, pp. 106 – 107, February 1974, Invited paper.

[2] R. W. Brodersen *et al.*, "Methods for true power minimization," in *IEEE/ACM International Conference on Computer Aided Design*, November 2002, pp. 35 – 42.

[3] R. Gonzalez, B. M. Gordon, and M. A. Horowitz, "Supply and threshold voltage scaling for low power CMOS," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 8, pp. 1210 – 1216, August 1997.

[4] J. Tschanz *et al.*, "Dynamic sleep transistor and body bias for active leakage power control of microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 1838 – 1845, November 2003.

[5] International Technology Roadmap for Semiconductors. `http://public.itrs.net/`.

[6] S. R. Nassif, "Design for variability in DSM technologies," in *IEEE International Symposium on Quality Electronic Design*, March 2000.

[7] K. T. Ulrich and S. D. Eppinger, *Product Design and Development*, 3rd ed. McGraw Hill, 2004.

[8] M. Eisele, J. Berthold, D. Schmitt-Landsiedel, and R. Mahnkopf, "The impact of intra-die device parameter variations on path delays and on the design for yield of low voltage digital circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 5, no. 4, pp. 360 – 368, December 1997.

[9] P. A. Stolk, F. P. Widdershoven, and D. B. M. Klaassen, "Modeling statistical dopant fluctuations in MOS transistors," *IEEE Transactions on Electron Devices*, vol. 45, no. 9, pp. 1960 – 1971, September 1998.

[10] R. Zimmerman and W. Fichtner, "Low-power logic styles: CMOS versus pass transistor logic," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 7, pp. 1079 – 90, July 1997.

[11] S. R. Nassif, "Delay variability: Sources, impacts, and trends," in *Proceedings from the IEEE International Solid-State Circuits Conference*, February 2000.

[12] Y. Cao *et al.*, "Design sensitivities to variability: extrapolations and assessments in nanometer VLSI," in *Proceedings from the IEEE International ASIC/SoC Conference*, September 2002, pp. 411 – 415.

[13] D. J. Frank, P. Solomon, S. Reynolds, and J. Shin, "Supply and threshold voltage optimization for low power design," in *International Symposium on Low Power Electronics and Design*, 1997, pp. 317 – 322.

[14] BSIM3 Device Model. `http://www-device.eecs.berkeley.edu/~bsim3/`.

[15] Berkeley Predictive Technology Model. `http://www-device.eecs.berkeley.edu/~ptm/`.

[16] S. Dhar and M. Franklin, "Optimum buffer circuits for driving long uniform lines," *IEEE Journal of Solid-State Circuits*, vol. 26, no. 1, pp. 32 – 40, January 1991.

[17] N. Weste and K. Eshragian, *Principles of CMOS VLSI Design: A Systems Perspective*, 2nd ed. Addison-Wesley, 1993.

[18] E. B. Fowlkes, *A Folio of Distributions*. Marcel Dekker Inc., 1987.

[19] D. Liu and C. Svensson, "Trading speed for low power by choice of supply and threshold voltages," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 1, pp. 10 – 17, January 1993.

[20] Y. Cao *et al.*, "Yield optimization with energy-delay constraints in low power digital circuits," in *IEEE Conference on Electron Devices and Solid-State Circuits*, December 2003, pp. 285 – 288.

[21] P. Friedberg *et al.*, "Modeling within-die spatial correlation effects for process-design co-optimization," to be presented at the *IEEE International Symposium on Quality Electronic Design*, March 2005.

[22] J. D. Meindl and J. A. Davis, "*The* fundamental limit on binary switching energy for terascale integration (TSI)," *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 1515 – 1516, October 2000.

[23] K. Bernstein and N. Rohrer, *SOI Circuit Design Concepts*. Kluwer Academic Publishers, 2000.

[24] BSIM SOI Device Model. `http://www-device.eecs.berkeley.edu/~bsimsoi/`.

[25] C.-L. Chen and G. S. Ditlow, "Pulsed static CMOS circuit," US Patent No. 05495188, February 1996.

[26] K. Bernstein, "Basic logic families," in *Design of High-Performance Microprocessor Circuits*, A. Chandrakasan, W. J. Bowhill, and F. Fox, Eds. IEEE Press, 2001, ch. 7, pp. 119 - 139.

[27] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Prentice Hall, 2003.

[28] P. Kogge and H. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Transactions On Computers*, 1973.

[29] T. Han and D. Carlson, "Fast area-efficient VLSI adders," in *8th Annual Symposium on Computer Arithmetic*. Como Italy, March 1982, pp. 49 – 56.

[30] R. Brent and H. Kung, "A regular layout for parallel adders," *IEEE Transactions on Computers*, 1982.

[31] J. Tschanz *et al.*, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 1396 – 1402, November 2002.

[32] R. Hegde and N. Shanbhag, "A voltage overscaled low-power digital filter IC," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 2, pp. 388 – 391, February 2004.

[33] S. Niranjan and J. Frenzel, "A comparison of fault-tolerant state machine architectures for space-borne electronics," *IEEE Transactions on Reliability*, vol. 45, pp. 109 – 113, March 1996.

[34] S. P. Khatri *et al.*, "Engineering change in a non-deterministic FSM setting," in *Proceedings of the Conference on Design Automation*, June 1996, pp. 451 – 456.

[35] PicoRadio Project at Berkeley Wireless Research Center. `http://bwrc.eecs.berkeley.edu/Research/Pico_Radio/`.

[36] MVSIS Project Page. `http://www-cad.eecs.berkeley.edu/mvsis/`.

[37] Berkeley Logic Interchange Format (BLIF). University of California Berkeley, July 1992.