Robust Near-Threshold QDI Circuit Analysis and Design

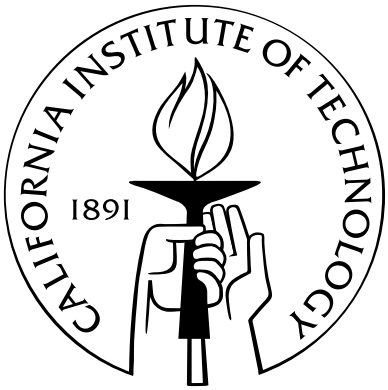
Thesis by

Sean Keller

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy



California Institute of Technology

Pasadena, California

2014

(Defended August 9, 2013)

Every day is a new day. It is better to be lucky.

But I would rather be exact. Then when luck comes you are ready.

—Ernest Hemingway - *The Old Man and the Sea*

To Meg.

Acknowledgements

There is an art to research; picking problems that are interesting, useful, and tractable falls somewhat outside the scope of the scientific method. It is a skill that probably cannot be directly taught, but rather it requires an immersive experience working alongside experts. I would like to thank my advisor Prof. Alain J. Martin for providing me with such an experience, and for teaching me many important lessons both directly and indirectly.

I am also grateful to Prof. David Money Harris for introducing me to the world of device modeling and for pushing me to develop a new MOS model for near-threshold circuits. I would also like to thank Dr. Michael Katelman for putting much time and energy into our collaborative project of applying formal-methods to asynchronous circuits; this work turned out to be the starting point for much of the research presented in this dissertation. I also need to thank a number of other current and former graduate students at Caltech: Siddharth S. Bhargav, Chris Moore, Xiaofei Chang, Nikil Mehta, Piyush Prakash, and Wonjin Jang. I would also like to thank my committee (Prof. Adam Wierman, Prof. Azita Emami-Neyestanak, Prof. David Money Harris, and Prof. Alain J. Martin) for taking the time to read and provide useful comments on my dissertation. Finally, I am thankful for the loving encouragement from my wife Meg, my parents, my grandmother, and my sister.

Abstract

The two most important digital-system design goals today are to reduce power consumption and to increase reliability. Reductions in power consumption improve battery life in the mobile space and reductions in energy lower operating costs in the datacenter. Increased robustness and reliability shorten down time, improve yield, and are invaluable in the context of safety-critical systems. While optimizing towards these two goals is important at all design levels, optimizations at the circuit level have the furthest reaching effects; they apply to all digital systems. This dissertation presents a study of robust minimum-energy digital circuit design and analysis. It introduces new device models, metrics, and methods of calculation—all necessary first steps towards building better systems—and demonstrates how to apply these techniques. It analyzes a fabricated *chip* (a full-custom QDI microcontroller designed at Caltech and taped-out in 40-nm silicon) by calculating the minimum energy operating point and quantifying the *chip's* robustness in the face of both timing and functional failures.

Contents

List of Figures

Chapter 1

Introduction

1.1 Challenges

Energy efficiency is one of the most important design concerns for modern digital CMOS integrated circuits and systems [4, 21]. Reducing energy consumption results in longer battery life for mobile devices, and energy savings in data centers translates directly to reduced operating costs. Energy can be reduced via system optimization at every level (*e.g.,* software, firmware, architecture, circuits). Among these different design levels, circuit optimizations have the furthest reaching effects; *i.e.,* circuit optimizations are applicable to all digital systems. To the first order, the energy consumed by a digital circuit is quadratic in the supply voltage (V_{DD}), so reducing V_{DD} plays a critical role in reducing energy. Nevertheless, due to second-order effects (*e.g.,* leakage currents) as V_{DD} is reduced, a minimum-energy (per-cycle) operating point is reached at a non-zero supply voltage.

In order to build minimal-energy systems, the minimum-energy supply voltage must first be determined. The technology employed, the structure and size of the system, and external factors (*e.g.,* temperature) have a significant impact on the minimum-energy operating point; however, the operating point is typically near the device threshold voltage (V_t) [22, 34, 42, 52]. For this reason, there is considerable interest in the analysis of near-threshold circuit operation. This in turn necessitates accurate (and usable) near-threshold device models; however, current models are either inaccurate, discontinuous around V_t, or too cumbersome to use.

Problematically, lowering the supply voltage to near-threshold exponentially reduces system reliability (compared to reliability at the process nominal V_{DD}); this reliability reduction is largely attributable to parameter variation. Modern MOS manufacture is certainly among the most sophisticated industrial processes ever developed and successfully employed, but the process is imperfect, and devices are so small and numerous that atomistic effects are unavoidable. These process imperfections and atomistic effects yield devices with physical parameters that vary stochastically (*i.e.,* parameter variation). A design optimized for the typical case may fail because one or more devices are significantly skewed from their nominal value. Parameter variation-induced failures can be classified as either timing failures or functional failures. An assumption about a path delay (timing assumption) that is nominally valid may not hold true in actual silicon because of

parameter variation (a timing failure). A gate may simply fail to switch, or a memory may fail to hold state (a functional failure). The classic engineering approach to this problem, using worst-case design margins, is no longer practical or efficient [18].

The design of near-threshold circuits that function reliably despite parameter variation requires new models and techniques for analysis. Near-threshold statistical delay models must be developed and verified so that timing assumptions can be verified and their probability of failure estimated. Furthermore, no universal means of calculating statistical circuit robustness currently exists. To build reliable near-threshold circuits, the corresponding robustness must be quantified; a new metric and method of calculation is needed. Finally, QDI (quasi-delay insensitive) asynchronous circuits are widely considered to be the most robust family of digital circuits [58]. To that end, they offer an inherently practical solution to the problems of parameter variation and near-threshold circuit operation.

QDI circuits form an elegant class of clockless circuits that rely on a single critical design assumption called the isochronic fork assumption, a *necessary* [60] constraint on the relative path delays through branches of some forks. In order to calculate the probability of a timing failure in a QDI system, the exact timing constraint imposed by the isochronic fork assumption needs to be determined. With an accurate near-threshold device model, the minimum energy operating point for a QDI circuit can be determined. With a statistical delay model, the probability of timing failures in a QDI circuit can be computed, and with a robustness metric, the probability of functional failures can be calculated. Accurate interconnect models and manufacturing defect models already exist, and so the robustness and yield of a QDI design operating near-threshold can be specified.

1.2 Contributions

This dissertation sets out to address the hurdles detailed in Section 1.1. Chapter 2 presents and verifies the near-threshold model, a physically derived transregional MOS model. This new model is then used to generate a near-threshold statistical delay model. Chapter 3 introduces a new metric for statistical circuit robustness and an efficient means of calculation. Chapter 4 analyzes the isochronic fork assumption in detail, and presents and formally proves the necessary and sufficient timing assumption for QDI circuits. Finally, in Chapter 5 these methods are used to analyze a modern (40-nm) QDI microcontroller designed at Caltech; the minimum energy operating point is determined and the probability of failure due to both timing failures and functional failures is estimated.

1.3 Collaboration

The work presented in this dissertation is mine, but as with most research it stems from a number of different collaborations. The work presented in this Chapter 2 comes from a rich collaboration with Prof. David

Money Harris of Harvey Mudd College. Prof. Harris proposed a new empirical model (in [42]), which was the impetus for the near-threshold model. A version of this chapter has been submitted for publication. The work presented in Chapter 3 stems from collaboration with Siddharth S. Bhargav and Chris Moore. Some of this work was presented at the 2nd European Workshop on CMOS Variability [52]. A journal version of this chapter is in preparation for submission to the IEEE Transactions on VLSI. Finally, Chapter 4 stems from a fruitful collaboration with Dr. Michael Katelman during his doctoral work under Prof. José Meseguer at the University of Illinois at Urbana-Champaign. A version of this work was presented at the 15th IEEE Symposium on Asynchronous Circuits and Systems.[51]. Dr. Katelman later extended and formally verified much of this work [49, 50].

1.4 Thesis Statement

The minimum-energy operating point of modern nanoscale digital CMOS systems can be accurately modeled and occurs near the device threshold voltage. Parameter variation decreases the robustness of circuits operating near-threshold, but a metric to quantify statistical robustness can be developed along with an efficient method of calculation. Lastly, robust quasi-insensitive (QDI) systems can be designed to operate near-threshold, and the probability of timing and functional failures can be calculated.

Chapter 2

A Compact Transregional Model for Digital CMOS Circuits Operating Near-Threshold

2.1 Introduction

Power and energy dissipation are critical design constraints in modern digital systems. Minimizing power and energy consumption in CMOS—the dominant digital circuit technology—requires supply voltage scaling below the process nominal supply voltage (V_{DD}). The minimum energy operating point can occur below the device threshold voltage (V_t) or above it and is a function of process parameters and environmental factors (such as activity factor) [21, 57, 96, 97]. Even with additional constraints (*e.g.,* performance, reliability, yield), the energy optimal operating point typically occurs near the threshold voltage [22, 34, 42, 52]. For these reasons, there is considerable interest in the analysis of circuits operating near-threshold.

Modeling and analysis in this region of interest, around the threshold voltage, is complicated by the fact that even a rather narrow range of a few hundred millivolts around V_t spans three distinct MOSFET operating regimes: weak inversion, moderate inversion, and strong inversion. Conventional compact digital MOSFET models—the linear/quadratic strong inversion model [85], the alpha-power law model [17, 77], and the exponential weak inversion model [85]—are discontinuous and inaccurate around V_t. Accurate continuous models exist [91], and some have been applied to digital circuit analysis. Nevertheless, it is apparent from [57] that even the simplest of continuous models are difficult to work with and yield complicated expressions for digital circuits (*e.g.,* delay and energy) that somewhat obscure the relationship to supply voltage.[1] This relational complexity speaks to a clear need for MOS models that are simple enough to work with and reason about, while being sufficiently accurate to yield usable results. One of the goals of this chapter is to address this problem; that is, to clarify the energy and delay relationship to the supply voltage (near-threshold) by deriving a new simplified drain-current model.

[1] Markovic et al. are aware of this complexity and do acknowledge it in [57].

Compact MOS models are usually developed to be used in conjunction with numerical solvers and circuit simulators, as opposed to being designed for hand calculations. The most accurate of these models tend to have the greatest computational complexity and are the most difficult to work with by hand, while the simplest have reduced computational complexity at the expense of accuracy. Circuit simulation, along with the associated models, certainly plays an important role in digital system design; however, simple models and hand analysis can give the designer deeper insight into key trade-offs, potential circuit problems, and optimizations than can be achieved by simulation alone. This chapter presents a MOS device model designed specifically for hand calculations involving digital circuits.

Toward the goal of reducing model complexity, a number of simplifications are made throughout this chapter. One such simplification reduces the drain-current (I_{ds}) model to a *digital* current model. In digital circuit design, first-order approximations for important characteristics (*e.g.,* energy and delay) of large gate networks require only two MOSFET models: (1) the drain current of a logically "on" transistor (I_{on}) as a function of V_{DD}, and (2) the drain current of a logically "off" transistor (I_{off}). Simple but accurate models for I_{off} exist, and those that include short channel effects are adequate. On the other hand, there is a need for new I_{on} models that are accurate across all operating regimes. Of course, using I_{on} and I_{off} in lieu of a general I_{ds} model eliminates a number of variables and reduces model complexity but is only appropriate for *digital* applications.

This chapter presents a simple, physically derived, inverted-charge MOS device model for I_{on} (Equation 2.41) that is accurate for supply voltages ranging from a few times the thermal voltage to approximately twice the threshold voltage in modern technologies; *i.e.,* it is transregional. Since this model is approximately centered at V_t, it is referred to throughout as the near-threshold model. The model is continuous and continuous in the first derivative; it makes use of three process-independent fitting parameters, and these parameters are *stable*. That is, these fitting parameters remain constant and the model remains accurate across different process technologies. Moreover, the model is shown to be accurate across four different commercial technologies from two different foundries ranging from 40 nm to 90 nm. The organization of the remainder of this chapter is as follows. Section 2.2 gives the derivation of the near-threshold model. Section 2.3 applies the model derived in Section 2.2 to several problems. Section 2.4 discusses related work, and Section 2.5 concludes the chapter.

2.2 A Compact Near-Threshold I_{on} Model

It is tempting to avoid the considerable trouble of developing a physical model, and rather to use an empirical curve-fit as the foundation for a simplified transregional model. The problem with a purely empirical approach—even if the model is only intended for digital circuit analysis—is twofold. First, it is difficult to stabilize the model with respect to physical parameters that vary, such as the threshold voltage. Second, it is difficult to trust such a model; it is not clear how the fitting constants might change in new or differ-

ent technologies. Fortunately, there are a number of established physical MOS models and approaches to compact modeling. One such approach, inversion-charge modeling, is used in this chapter to generate the near-threshold model.

Inversion-charge models differ from the classic surface-potential-based models in that they make explicit the relation between MOS terminal voltages and the inversion-charge density (*e.g.,* the charge due to electrons below the gate of an NFET). A continuous expression for drain current as a function of terminal voltages follows directly from this explicit relation when applied to the Pao-Sah [72] model. The inversion-charge density to terminal voltage relation is difficult to compactly model, and the choice of simplifying approximations is a key differentiating factor between inversion-charge models.

The goal of this section is to derive a new analytical expression for NFET drain current of an "on" transistor, I_{on}, where I_{on} is defined as the drain current when $V_{gb} = V_{db} = V_{DD}$ and $V_{sb} = 0V$. This expression for I_{on} and the derivation are also applicable to a PFET; however, the corresponding derivation is not presented. The derivation begins with the quasi-static long-channel model for an NFET in terms of gate, source, and drain voltages, all relative to the bulk along the lines of the EKV model derivation presented in [37]; as such, some of the content presented in Section 2.2.1 and 2.2.2 is a review. It is a long/wide channel inversion-charge model that makes use of the linearization of inversion-charge to surface potential. The derivation starts with a well-accepted expression for drain current in terms of a diffusion component and a drift component, which is reduced to an expression where the drain current is proportional to a one-dimensional integral from the source potential to the drain potential of the mobile inversion-charge (*i.e.,* electrons) in the channel. A number of normalizations are applied to simplify the expression, and the integral is broken into an equivalent difference expression. Next, an expression is given for the mobile inversion-charge in terms of the normalized gate, source, drain, and threshold voltages. This expression is directly solved for mobile inversion-charge without approximation, a task that previous works were unable to accomplish. Additionally, several approximations are explained, and a new approximation that yields the near-threshold model is presented. These approximations for mobile inversion-charge can be directly applied to the integral expression for drain current to give drain current in terms of the terminal voltages. Finally, this drain-current expression is further simplified to give I_{on} as a function of V_{DD}.

2.2.1 Drain-Current Model

Consider an NFET labeled as in Figure 2.1. The standard long/wide-channel expression for drain current is given by Equation 2.1. See [40, 91] for a full derivation and a discussion of the physical assumptions required for validity.

$$I_{ds}(x) = \mu W \left(-Q_i' \frac{d\psi_s}{dx} + \phi_t \frac{dQ_i'}{dx} \right), \tag{2.1}$$

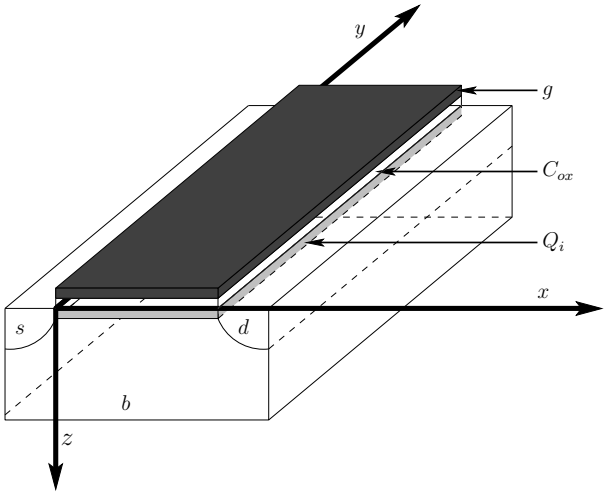


Figure 2.1: NFET physical view.

where μ is the effective electron mobility, W is the channel width, $Q'_i(x)$ is the mobile inversion-charge per unit area as a function of position along the channel, ϕ_t is the thermal voltage[2], and $\psi_s(x)$ is surface potential (the potential drop from the semiconductor surface to deep into the body). The first term, $-Q'_i \frac{d\psi_s}{dx}$, models *drift* and the second term, $\phi_t \frac{dQ'_i}{dx}$, models *diffusion*.

Assuming a constant channel width, a constant electron mobility, the charge sheet approximation (the entire mobile inversion-charge is at the surface potential), and the gradual channel approximation (the electric field along the z-axis is much larger than the field along the x-axis), Equation 2.1 reduces to Equation 2.2; see [37, 40] for a full discussion.

$$I_{ds} = \mu C'_{ox} \frac{W}{L} \int_{V_s}^{V_d} \frac{-Q'_i}{C'_{ox}} dV_c, \tag{2.2}$$

where C'_{ox} is the oxide capacitance per unit area, L is the channel length, and $V_c(x)$, the channel potential, represents the quasi-Fermi potential of electrons in the channel as a function of position; to the first order, it varies monotonically from the source to the drain, *i.e.,* from V_s to V_d.[3]

For a fixed V_g and V_s, as V_d, increases the device eventually enters the *saturation region*. This is due to the drain-end of the channel *pinching-off* as the drain-end enters weak-inversion and the mobile inversion-charge becomes negligible. Intuitively, this happens anywhere along the channel where the channel voltage is sufficiently large. In general, this property can be stated as the assumption that

$$\lim_{V_c \to \infty} Q'_i = 0. \tag{2.3}$$

As such, Equation 2.2 can be broken into two pieces: a forward current, I_f, which is independent of V_d, and

[2]Note that $\phi_t = \frac{k_B T}{q}$, where k_B is the Boltzmann constant, T is the absolute temperature, and q is the magnitude of the electrical charge on the electron.

[3]Terminal voltages are body referenced unless otherwise specified.

a reverse current, I_r, which is independent of V_s. That is,

$$I_{ds} = \underbrace{\mu C'_{ox}\frac{W}{L}\int_{V_s}^{\infty}\frac{-Q'_i}{C'_{ox}}dV_c}_{I_f} - \underbrace{\mu C'_{ox}\frac{W}{L}\int_{V_d}^{\infty}\frac{-Q'_i}{C'_{ox}}dV_c}_{I_r}. \tag{2.4}$$

In order to solve this integral, the relationship between the channel potential and the mobile inversion-charge needs to be established; however, the precise relation is quite complicated. One simplifying approach is to assume a linear relationship between the mobile inversion-charge and the surface potential. This greatly simplifies the problem and yields a constant of proportionality, n, the slope factor. From [36],

$$n = \frac{Q'_i}{C'_{ox}(\psi_s - \psi_p)}, \tag{2.5}$$

where ψ_p is the pinch-off surface potential, the surface potential at which the inversion charge becomes zero.

Before solving Equation 2.4, it is convenient to normalize the terms to unit-less quantities using

$$\begin{aligned} q_i &= \frac{-Q'_i}{2n\phi_t C'_{ox}}, & I_0 &= 2n\mu C'_{ox}\frac{W}{L}\phi_t^2, \\ v_c &= \frac{V_c}{\phi_t}, & i &= \frac{I}{I_0}, \\ \frac{V_s}{v_s} &= \frac{V_d}{v_d} = \frac{V_g}{v_g} = \phi_t, & \text{and} \quad \frac{I_{ds}}{i_{ds}} &= \frac{I_f}{i_f} = \frac{I_r}{i_r} = I_0. \end{aligned} \tag{2.6}$$

Equation 2.2 now simplifies to

$$i_{ds} = \int_{v_s}^{v_d} q_i dv_c, \tag{2.7}$$

and Equation 2.4 becomes

$$i_{ds} = \underbrace{\int_{v_s}^{\infty} q_i dv_c}_{i_f} - \underbrace{\int_{v_d}^{\infty} q_i dv_c}_{i_r}. \tag{2.8}$$

Since the model is symmetric with respect to the source and drain, the forward and reverse component are of the same form; it is convenient to use a *combined* notation so as to work with both expressions (i_f and i_r) simultaneously. That is,

$$i_{f,r} = \int_{v_{s,d}}^{\infty} q_i dv_c. \tag{2.9}$$

Finally, all that is needed to solve Equation 2.9 is an expression for q_i, thus yielding an expression for drain current in terms of the three transistor terminal voltages - a goal of this section.

In normalized terms, the relation between mobile inversion-charge density and channel potential can be expressed as (see [37] for details)

$$2q_i + lnq_i = v_p - v_c, \tag{2.10}$$

where $v_p = \frac{V_p}{\phi_t}$ is the *pinch-off voltage*, defined in [79, 78] as

$$v_p = \psi_p - \psi_0, \tag{2.11}$$

where ψ_0 is a process-dependent term with various approximations used in the literature. Conveniently, v_p can be approximated with common terms as

$$v_p \approx \frac{V_g - V_t}{n\phi_t}, \tag{2.12}$$

where V_t is the *threshold voltage* [36, 79].

Equations 2.10 and 2.12 give the relation between the gate and channel potential *and* the mobile inversion-charge with the process dependent component compacted into the definition of v_p. Different [40] and more accurate [78] relations exist, but Equation 2.10 is simple, practical, and differentiable:

$$dv_c = -dq_i(2 + \frac{1}{q_i}). \tag{2.13}$$

Substituting this expression for dv_c into Equation 2.9 and integrating results in

$$i_{f,r} = q_{s,d}^2 + q_{s,d}, \tag{2.14}$$

where q_s is the normalized mobile inversion-charge at the source-end of the channel, and similarly for q_d at the drain end. Applying Equation 2.10 to the source and drain ends of channel yields

$$v_p - v_{s,d} = 2q_{s,d} + lnq_{s,d}. \tag{2.15}$$

Prior work (*e.g.,* [37, 78]) assumed that Equation 2.15 (and Equation 2.10) is not invertible, but it actually can be inverted by using the principal branch of the Lambert W function. The Lambert W function is defined as the root of

$$\mathsf{W}(z)e^{\mathsf{W}(z)} = z, \tag{2.16}$$

for any complex number z, (see [29] for details). The function dates back to the days of Euler and has been recently used in several related works (see Section 2.4).

After exponentiation, Equation 2.15 can be rearranged as

$$2q_{s,d}e^{2q_{s,d}} = 2e^{v_p - v_{s,d}}. \tag{2.17}$$

Table 2.1: Operating regime bounds for I_{on}

Operating Regime	Current Bounds	Potential Bounds
Weak Inversion	$i_f < -1.4$	$v_p - v_s < 0.20$
Moderate Inversion	$-1.4 \leq i_f < 3.6$	$0.20 \leq v_p - v_s < 4.0$
Strong Inversion	$3.6 \leq i_f$	$4.0 \leq v_p - v_s$

Applying the Lambert W function[4] to Equation 2.17 gives the closed-form expression

$$q_{s,d} = \frac{\mathsf{W}_0\left(2e^{v_p - v_{s,d}}\right)}{2}. \tag{2.18}$$

Analogously, applying the Lambert W function to Equation 2.10 gives the closed-form expression

$$q_i = \frac{\mathsf{W}_0\left(2e^{v_p - v_c}\right)}{2}, \tag{2.19}$$

(depicted in Figure 2.2(a)). This expression for q_i proves useful for making approximations in Section 2.2.3.

Finally, Equation 2.18 can be directly applied to Equation 2.14, giving a new closed-form expression for normalized drain current

$$i_{f,r} = \left(\frac{\mathsf{W}_0\left(2e^{v_p - v_{s,d}}\right)}{2}\right)^2 + \frac{\mathsf{W}_0\left(2e^{v_p - v_{s,d}}\right)}{2}. \tag{2.20}$$

This expression for $i_{f,r}$ is exact, while the EKV approximation [37], discussed in Section 2.2.2 and given by Equation 2.28, has a maximum absolute error of 21%. Using a more accurate approximation for inversion charge, *e.g.,* [78], and then using the Lambert W function to give an exact expression for inversion charge may further improve total model accuracy; however, this analysis falls outside of the scope of this dissertation and is left as future work.

Figure 2.3(a) depicts Equation 2.20, and it makes clear the nonlinear nature of drain current as a function of the terminal voltages. It also helps relate $i_{f,r}$ to the standard operating regimes: weak, moderate, and strong inversion. The model presented in this chapter is symmetric with respect to the source and drain; however, the ultimate goal of this derivation is to generate a model for I_{on} wherein the drain-end of the channel is tied to V_{DD} and the source-end to the body. From this and Equation 2.15, it follows that $q_s > q_d$,[5] *i.e.,* the inversion charge density at the source-end of the channel always exceeds that of the drain-end. From Equation 2.14 it follows that $i_f > i_r$, and so the operating regime is determined exclusively by i_f and correspondingly $v_p - v_s$. The drain-end of the channel, and the drain-dependent current i_r, are pinned in the weak-inversion regime. The boundaries between operating regimes are approximated in Table 2.1.[6] It should be noted that in the general case, the operating regime can be determined by the larger of i_f or i_r.

[4] When the domain of the Lambert W function is restricted to the non-negative reals, the co-domain reduces to that of the reals, and $\mathsf{W}(z)$ has a single value denoted by the principal branch $\mathsf{W}_0(z)$.

[5] This requires that V_{DD} is a positive value relative to the body.

[6] Analytical bounds on operating regimes can be found in [91].

2.2.2 Existing Drain-Current Approximations

There are three well accepted approximations for $i_{f,r}$, a simple weak-inversion approximation, a simple strong-inversion approximation, and a continuous approximation which is valid in all operating regions. The weak- and strong-inversion approximation, along with the new near-threshold model, are generated by modeling the mobile inversion-charge as a function of the terminal voltages; Figure 2.2 and Figure 2.3 graphically depict these charge and current approximations respectively.

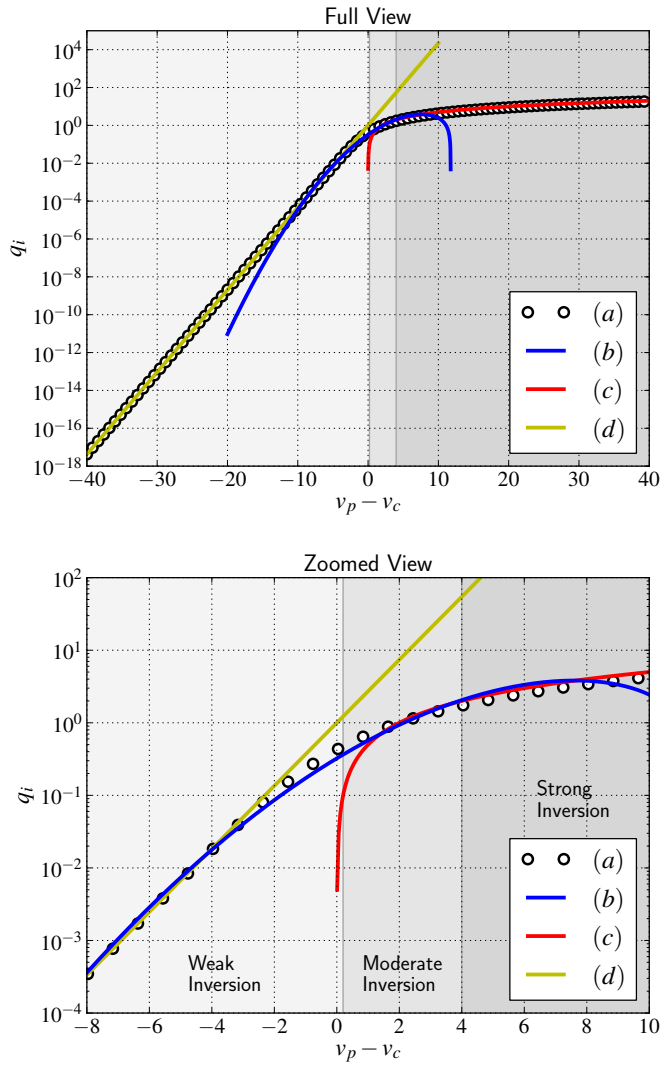


Figure 2.2: (a) Equation 2.19 (b) Equation 2.35 (Near-Threshold Model) (c) Equation 2.26 (Strong Inversion Approximation) (d) Equation 2.22 (Weak Inversion Approximation).

In weak-inversion, $v_p - v_c \ll 0$, and from Equation 2.10 it follows that $2q_i + lnq_i \ll 0$. The logarithmic term dominates, so

$$v_p - v_c \approx lnq_i, \text{ and} \tag{2.21}$$

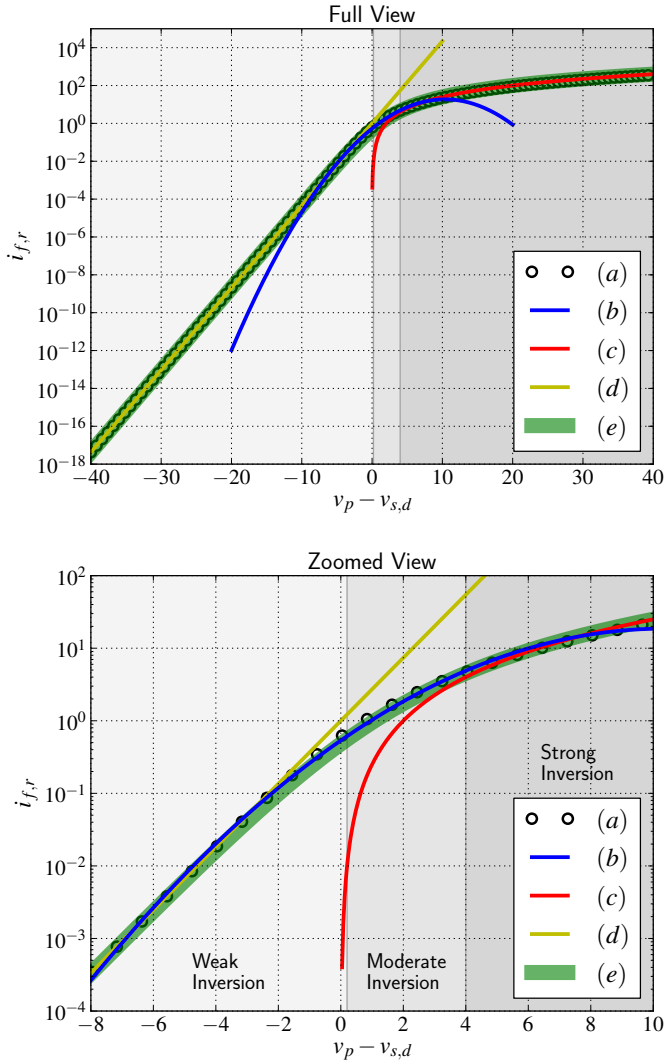


Figure 2.3: (a) Equation 2.20 (b) Equation 2.36 (Near-Threshold Approximation) (c) Equation 2.27 (Strong Inversion Approximation) (d) Equation 2.23 (Weak Inversion Approximation) (e) Equation 2.28 (EKV Continuous Approximation).

$$q_i \approx e^{v_p - v_c} \tag{2.22}$$

(see Figure 2.2(d)). Integrating Equation 2.9 with this approximation gives

$$i_{f,r} \approx e^{v_p - v_{s,d}}, \tag{2.23}$$

depicted in Figure 2.3(d). Removing the normalization, letting the approximation become an equality, and combining the forward and reverse components yields a well-known equation for sub-threshold drain current

$$I_{ds} = I_0 e^{\frac{V_g - V_t}{n\phi_t}} \left(e^{\frac{-V_s}{\phi_t}} - e^{\frac{-V_d}{\phi_t}} \right). \tag{2.24}$$

In strong-inversion, $v_p - v_c \gg 0$, so the logarithmic term in Equation 2.10 is negligible. That is

$$v_p - v_c \approx 2q_i, \text{ and} \tag{2.25}$$

$$q_i \approx \frac{v_p - v_c}{2} \tag{2.26}$$

(see Figure 2.2(c)). With Equation 2.9,

$$i_{f,r} \approx \left(\frac{v_p - v_{s,d}}{2} \right)^2, \tag{2.27}$$

depicted in Figure 2.3(c).

Finally, the continuous EKV approximation [37] given by

$$i_{f,r} \approx ln^2 \left[1 + e^{\frac{v_p - v_{s,d}}{2}} \right], \tag{2.28}$$

as depicted in Figure 2.3(e), is accurate over all operating regimes (at the expense of increased complexity).

2.2.3 Transregional Near-Threshold Drain-Current Approximation

This subsection presents a new inversion-charge approximation and corresponding drain current approximation for digital circuits. The model is simpler than the EKV model and continuously models digital devices operating across weak, moderate, and strong inversion. Consider Equation 2.10; in weak-inversion the logarithmic terms dominates and in strong-inversion the linear term dominates. In moderate inversion neither term dominates, so a simple approximation is not possible. Fortunately, the exact expression for charge, Equation 2.19, can be simplified for a narrow range of $v_p - v_c$.

First, assuming that $q_i > 0$, taking the logarithm of both sides of Equation 2.19 gives

$$lnq_i = ln\mathsf{W}_0 \left(2e^{v_p - v_c} \right) - ln2. \tag{2.29}$$

Next, from [94], for $x > 0$ and $\mathsf{W}_0(x) > 0$

$$ln\mathsf{W}_0(x) = lnx - \mathsf{W}_0(x). \tag{2.30}$$

As such, Equation 2.29 can be expressed as

$$lnq_i = v_p - v_c - \mathsf{W}_0(2e^{v_p - v_c}). \tag{2.31}$$

Next, [29] shows that $\mathsf{W}(e^x)$ can be approximated by a Taylor series expansion. As such, Equation 2.31 can be written as

$$lnq_i \approx v_p - v_c - P(v_p - v_c), \tag{2.32}$$

for some polynomial P, wherein the coefficients and validity range are a function of $v_p - v_c$. The optimal polynomial approximation for a particular range of interest can be calculated to a high degree, but this does not aid in the simplification of the problem at hand. The approach taken in this chapter is to use a degree-two polynomial and to curve fit the entire expression. That is,

$$lnq_i \approx k_a + k_b(v_p - v_c) + k_c(v_p - v_c)^2, \tag{2.33}$$

where k_a, k_b, and k_c are fitting constants.

Finally, letting $k_f = e^{k_a}$, $v_\omega = v_p - v_c$, and exponentiating both sides of Equation 2.33 gives

$$q_i \approx k_f e^{k_b v_\omega + k_c v_\omega^2}. \tag{2.34}$$

In order to calculate $i_{f,r}$, integration is necessary, so it is helpful to approximate Equation 2.34 as

$$q_i \approx k_0(k_1 + 2k_2 v_\omega)e^{k_1 v_\omega + k_2 v_\omega^2}, \tag{2.35}$$

where k_0, k_1, and k_2 are new fitting constants.[7] (See Figure 2.2(b) for a graphical depiction.)

After substituting Equation 2.35 into Equation 2.9 and integrating, the resulting expression for normalized drain current can be expressed as

$$\mathbf{i_{f,r}} \approx \mathbf{k_0 e^{k_1 v_\varpi + k_2 v_\varpi^2}}, \tag{2.36}$$

where $v_\varpi = v_p - v_{s,d}$. Fitting this expression in the near-threshold region, $-8 < v_p - v_{s,d} < 10$, (see Section 2.2.4 for boundary definition) gives the fitting constants given in Table 2.2 (used throughout this chapter). (See Figure 2.3(b) for a graphical depiction.) Note that due to the definition of the pinch-off voltage and the use of normalized variables, the fitting constants (k_0, k_1, k_2) are process independent.

[7] This is valid, because taking the logarithm of both sides of Equation 2.35 gives $lnq_i \approx lnk_0(k_1 + 2k_2 v_\omega) + k_1 v_\omega + k_2 v_\omega^2$. Using the first few terms of the Taylor series for the ln term on the RHS reduces the entire RHS to a polynomial with new coefficients, *i.e.,* $lnq_i \approx P(v_\omega)$. As such, removing high-order terms and exponentiating both sides gives Equation 2.34.

Table 2.2: Near-threshold model fitting constants - error reported for $i_{f,r}$ (Equation 2.36 compared to Equation 2.20)

	Value
k_0	5.4e-1
k_1	6.9e-1
k_2	-3.3e-2
Maximum Absolute Error	8.1%
Mean Absolute Error	21%

Finally, combining this expression for $i_{f,r}$ (Equation 2.36) with Equation 2.8 gives

$$i_{ds} = k_0 e^{k_1(v_p - v_s) + k_2(v_p - v_s)^2} - k_0 e^{k_1(v_p - v_d) + k_2(v_p - v_d)^2}. \tag{2.37}$$

Removing the normalization, and using Equation 2.12 to approximate v_p yields

$$I_{ds} = I_0 k_0 e^{k_1(\frac{V_g - V_t}{n\phi_t} - \frac{V_s}{\phi_t}) + k_2(\frac{V_g - V_t}{n\phi_t} - \frac{V_s}{\phi_t})^2} - I_0 k_0 e^{k_1(\frac{V_g - V_t}{n\phi_t} - \frac{V_d}{\phi_t}) + k_2(\frac{V_g - V_t}{n\phi_t} - \frac{V_d}{\phi_t})^2}. \tag{2.38}$$

Now, referencing all voltages to the source instead of the body and assuming that $V_{sb} = 0V$, gives,

$$I_{ds} = I_0 k_0 e^{k_1 \frac{V_{gs} - V_t}{n\phi_t} + k_2(\frac{V_{gs} - V_t}{n\phi_t})^2} \left(1 - e^{k_1 \frac{-V_{ds}}{\phi_t} + k_2 \frac{n^2 V_{ds}^2 - 2nV_{ds}V_{gs} + 2nV_{ds}V_t}{n^2\phi_t^2}}\right). \tag{2.39}$$

For I_{on}, $V_{DD} = V_{gs} = V_{ds}$, so

$$I_{\text{on}} = I_0 k_0 e^{k_1 \frac{V_{DD} - V_t}{n\phi_t} + k_2(\frac{V_{DD} - V_t}{n\phi_t})^2} \left(1 - e^{k_1 \frac{-V_{DD}}{\phi_t} + k_2 \frac{n^2 V_{DD}^2 - 2nV_{DD}^2 + 2nV_{DD}V_t}{n^2\phi_t^2}}\right). \tag{2.40}$$

Assuming that V_{DD} is both a few times larger than ϕ_t and less than twice the threshold voltage, allows the terms within parentheses to be approximated as unity, and letting $V_{DT} = V_{DD} - V_t$,

$$\mathbf{I_{on} = I_0 k_0 e^{k_1 \frac{V_{DT}}{n\phi_t} + k_2 \left(\frac{V_{DT}}{n\phi_t}\right)^2}}. \tag{2.41}$$

Equation 2.41 gives the drain current of a logically "on" transistor as a function of the supply voltage, the goal of this section and one of the main goals of this chapter. Within this expression, the constants k_0, k_1, and k_2 are process independent, and the process dependent terms are contained in the definitions of I_0, n, and V_t. The definition of I_0 (Equation 2.6) also contains the sizing ratio $\frac{W}{L}$. This ratio is intentionally kept within the definition of I_0 throughout, because in short/narrow-channel devices, modifying gate dimensions can affect some or all of the process dependent terms. As with most compact models, short/narrow-channel effects can

Table 2.3: Model validity regions (bounded by a maximum absolute error of 21%)

Approximation	min($v_p - v_{s,d}$)	max($v_p - v_{s,d}$)	min(i_f)	max(i_f)	Mean Absolute Error
EKV Continuous	< -40	> 40	$< 4.2e{-}18$	$> 3.6e2$	6.9%
Weak Inversion	< -40	-1.4	$< 4.2e{-}18$	0.20	0.59%
Strong Inversion	3.6	> 40	4.0	$> 3.6e2$	11%
Near-Threshold	-8.0	10	$3.4e{-}4$	23	8.1%

be included in the near-threshold model as needed.[8] Additionally, regions of validity for both W and L can be established before using the near-threshold model (or any compact model) to calculate drain current as a function of either term.

2.2.4 Near-Threshold Model Validation

Figure 2.3 depicts the different approximations for normalized drain current as a function of the transistor terminal voltages; it is clear that each approximation has a particular region of validity. The region boundaries are difficult to determine analytically but can be readily defined in terms of a maximum error. The original EKV approximation, Equation 2.28, has a maximum absolute error of 21% compared to the analytical drain-current expression given by Equation 2.20. The EKV approximation is a useful and well accepted model, so the corresponding maximum absolute error of 21% against Equation 2.20 can also be used as a validity bound for the other drain-current approximations. Table 2.3 provides the region boundaries in terms of both normalized voltages and currents, along with the mean absolute error.

The near-threshold model is further validated by application to four commercial bulk CMOS processes from two different foundries. Nominal devices, high-threshold transistors (HVT), and low-threshold transistors (LVT) are modeled in a 40-nm low-power (LP) technology, a 65-nm low-power technology, a 65-nm general-purpose (GP) technology, and a 90-nm general-purpose technology[9]. The foundry-provided BSIM4 models for each technology node are used as the basis for comparison and for parameter extraction. Parameter extraction is performed by way of a least-squares fit. This method of parameter extraction is common and convenient, but it has shortcomings. In simplified models, such as those presented in this chapter, the extracted parameters may not correspond to the physical parameters that they are intended to represent. This is especially true of parameters that are greatly impacted by short-channel effects, *e.g.,* V_t which is affected by drain-induced barrier lowering (DIBL) [90].

Figures 2.4 and 2.5 each overlay the near-threshold model on top of the corresponding BSIM4 simulation of the 40-nm LP and 65-nm GP technologies, respectively. In these figures, the near-threshold model is plotted for the entire V_{DD} range to make clear how the model deviates outside of its range of applicability. Table 2.4 gives the lower and upper bounds on model applicability along with error rates (relative to BSIM4

[8] See [91] for an example of incorporating short-channel effects into a strong-inversion model.

[9] The 90-nm technology only permits nominal and LVT devices, so 90-nm HVT devices are not modeled.

Table 2.4: Near-Threshold model compared to SPICE simulation of BSIM4 model for commercial technologies (at 70°C) – the circuit parameters V_t, I_0, and n are extracted from a least-squares fit against the corresponding BSIM4 simulation

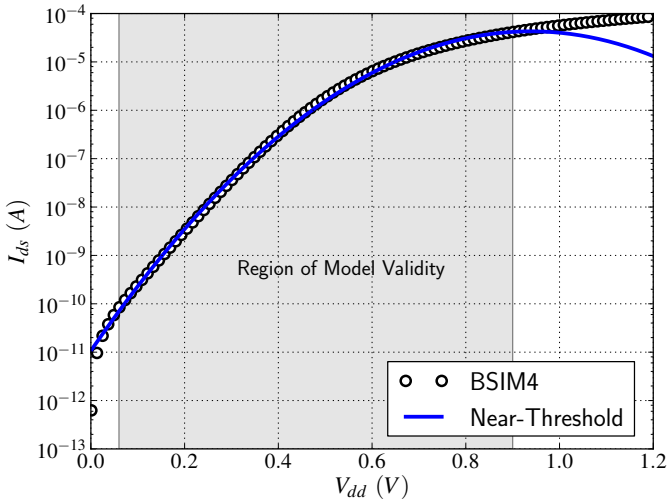
Technology	Device	V_t (mV)	L (nm)	W (nm)	$I_0(A)$	n	Maximum Absolute Error	Mean Absolute Error	Lower Bound (mV)	Upper Bound (mV)
40-nm Low Power	NFET	485	36	108	$2.30e{-6}$	1.48	15%	8.5%	60	900
40-nm Low Power	PFET	458	36	108	$1.10e{-6}$	1.24	13%	7.4%	60	900
40-nm Low Power	HVT NFET	586	36	108	$2.30e{-6}$	1.58	20%	12%	60	1100
40-nm Low Power	HVT PFET	572	36	108	$1.23e{-6}$	1.56	21%	13%	60	1100
40-nm Low Power	LVT NFET	415	36	108	$1.34e{-6}$	1.49	17%	8.3%	60	900
40-nm Low Power	LVT PFET	425	36	108	$2.67e{-6}$	1.45	16%	8.4%	60	900
65-nm General Purpose	NFET	310	50	100	$1.74e{-6}$	1.32	9.8%	4.9%	60	700
65-nm General Purpose	PFET	392	50	100	$8.99e{-7}$	1.34	12%	7.6%	60	700
65-nm General Purpose	HVT NFET	371	50	100	$1.83e{-6}$	1.34	12%	7.7%	60	700
65-nm General Purpose	HVT PFET	475	50	100	$1.11e{-6}$	1.50	11%	5.0%	60	700
65-nm General Purpose	LVT NFET	273	50	100	$2.22e{-6}$	1.33	13%	6.4%	60	700
65-nm General Purpose	LVT PFET	348	50	100	$8.76e{-7}$	1.32	9.3%	5.2%	60	700
65-nm Low Power	NFET	504	60	120	$1.86e{-6}$	1.46	19%	11%	60	900
65-nm Low Power	PFET	504	60	120	$1.12e{-6}$	1.50	16%	9.1%	60	900
65-nm Low Power	HVT NFET	610	60	120	$1.86e{-6}$	1.50	22%	13%	60	900
65-nm Low Power	HVT PFET	602	60	120	$1.25e{-6}$	1.59	21%	13%	60	900
65-nm Low Power	LVT NFET	373	60	120	$1.44e{-6}$	1.25	12%	6.3%	60	750
65-nm Low Power	LVT PFET	460	60	120	$1.28e{-6}$	1.52	11%	5.0%	60	750
90-nm General Purpose	NFET	304	80	120	$1.52e{-6}$	1.25	14%	7.0%	60	600
90-nm General Purpose	PFET	395	80	120	$1.24e{-6}$	1.30	15%	9.3%	60	600
90-nm General Purpose	LVT NFET	218	80	120	$1.24e{-6}$	1.18	21%	6.0%	60	600
90-nm General Purpose	LVT PFET	271	80	120	$3.62e{-7}$	1.20	6.2%	3.3%	60	600

simulation) and extracted parameter values. Table 2.4 also specifies the device dimensions and provides the data for LVT and HVT devices (where applicable). Note that the error associated with HVT devices tends to be greater than that of the corresponding regular devices. This can be attributed to modeling error at the low end of the V_{DD} range; that is, with HVT devices, the quantity $v_p - v_{s,d}$ can be less than the near-threshold model lower bound given in Table 2.3.

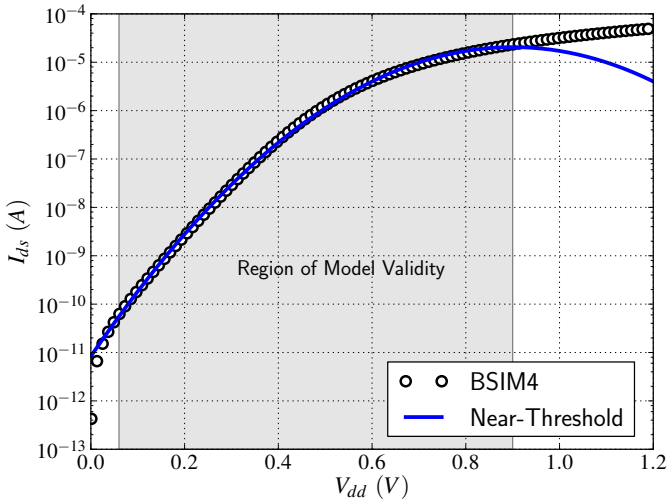
2.3 Near-Threshold Model Applications

The goal of this section is to demonstrate the applicability of the near-threshold model to digital circuit analysis in a modern technology. The model is first used to generate a closed-form analytical expression for delay, which is then used to give a closed-form equation for energy, and this is used to determine the minimum-energy operating point as a function of activity factor and frequency. Finally, parameter variation is incorporated into the model, and closed-form expressions for the stochastic path-delay are derived. All of these analyses, which yield closed-form expressions, leverage the simplicity of a digital I_{on} model designed for hand calculations.

Model validity is determined by comparing the analytical expressions against corresponding BSIM4 SPICE simulations, and the errors are reported. For simplicity, chains of minimum-size inverters are used as a basis throughout; minimum-size devices are typical of circuits designed to minimize energy in the near-

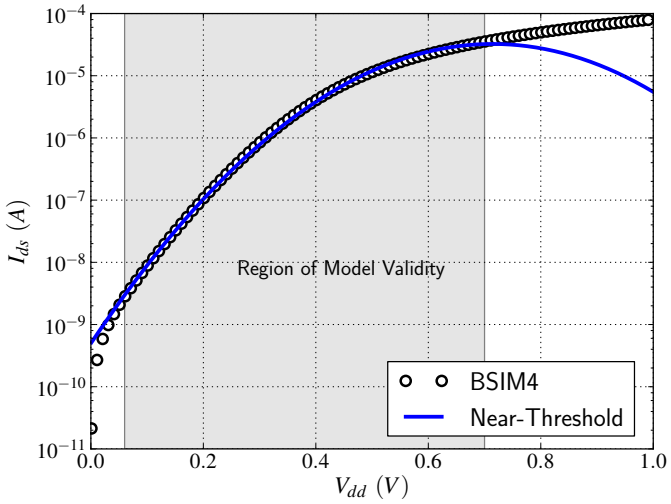


(a) NFET

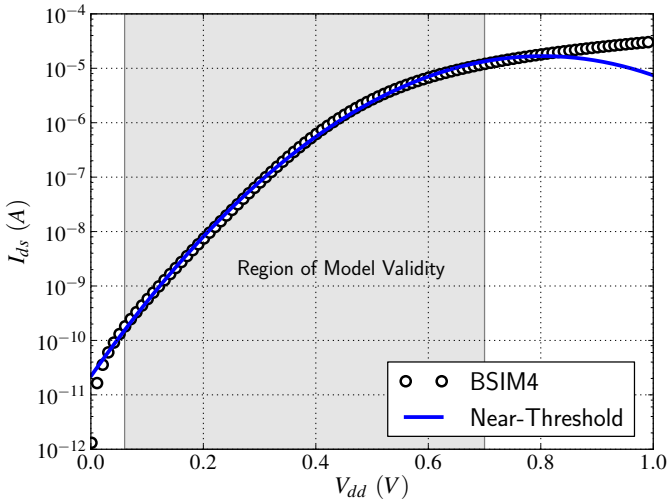


(b) PFET

Figure 2.4: Equation 2.36 (Near-Threshold Model) plotted for entire V_{DD} range against SPICE simulation of BSIM4 model of a 40-nm low-power process with minimum-size devices.



(a) NFET



(b) PFET

Figure 2.5: Equation 2.36 (Near-Threshold Model) plotted for entire V_{DD} range against SPICE simulation of BSIM4 model of a 65-nm general-purpose process with minimum-size devices.

threshold region. Chains of other gates can be normalized to this basis; the delays of more complex digital circuits, *e.g.,* a ripple-carry adder, track that of the inverter over a wide supply voltage range [42]. Similar analyses also make use of inverters as a canonical basis for analytical evaluation, *e.g.,* [21, 98]. Furthermore, Table 2.5 (in Section 2.3.1) reports the error (as compared to SPICE) when the closed-form delay model is applied to combinational gates other than minimum-size inverters.

2.3.1 Delay Model

Numerous delay models of varying accuracy and complexity have been used to model the switching delay of gates operating super-threshold, *e.g.,* [67, 75, 96]. For circuits operating sub-threshold and near-threshold, [21, 57, 42, 97] use and validate a simple linear RC-delay model. That is, the delay of a gate can be approximated as

$$t_{pd} = k_f C_{\text{load}} \frac{V_{DD}}{I_{\text{on}}}, \tag{2.42}$$

where C_{load} is the load capacitance, and k_f is a small fitting constant. This fitting constant serves to normalize the RC time constant and is necessary because propagation delay more closely tracks the drain current of devices that are only partially 'on' [67].

Using the near-threshold model for I_{on} (Equation 2.41), t_{pd} from Equation 2.42 can be expressed as

$$t_{pd} = \frac{k_f C_{\text{load}}}{k_0 I_0} V_{DD} e^{-k_1 \frac{V_{DT}}{n\phi_t} - k_2 \left(\frac{V_{DT}}{n\phi_t}\right)^2}. \tag{2.43}$$

Since I_0 is typically treated as a fitting constant, Equation 2.43 can be simplified by combining the constants k_f, k_0, and I_0 into a single term I_F. This gives

$$t_{pd} = \frac{C_{\text{load}}}{I_F} V_{DD} e^{-k_1 \frac{V_{DT}}{n\phi_t} - k_2 \left(\frac{V_{DT}}{n\phi_t}\right)^2}. \tag{2.44}$$

In order to apply Equation 2.44 to an inverter, separate delays for the PFET and NFET can be calculated. A simpler approach used in this chapter is to calculate an average propagation delay, which simultaneously models the delay of both types of devices, but this requires refitting I_0 and V_{DT}. Figure 2.6 plots the FO4 delay of a minimum-size inverter in the 65-nm GP process. The Near-Threshold model is plotted against the BSIM4 model with the Near-Threshold model fit from 135mV to 700mV (the inverters do not function below 135mV). The mean absolute error is 8.0%, and the maximum absolute error is 13% with $V_t = 386$mV, $n = 1.43$, and $\frac{C_{\text{load}}}{I_F} = 1.42\frac{ns}{V}$.

Equation 2.44 can be applied to a variety of gates by fitting $\frac{C_{\text{load}}}{I_F}$; Table 2.5 gives the corresponding error (as compared to the BSIM4 model) for the FO4 delay of several combinational gates: a minimum-size inverter, a four-times minimum-width inverter, an eight-times minimum-width inverter, a NAND2 gate, a NOR2 gate, and an AOI21 gate.

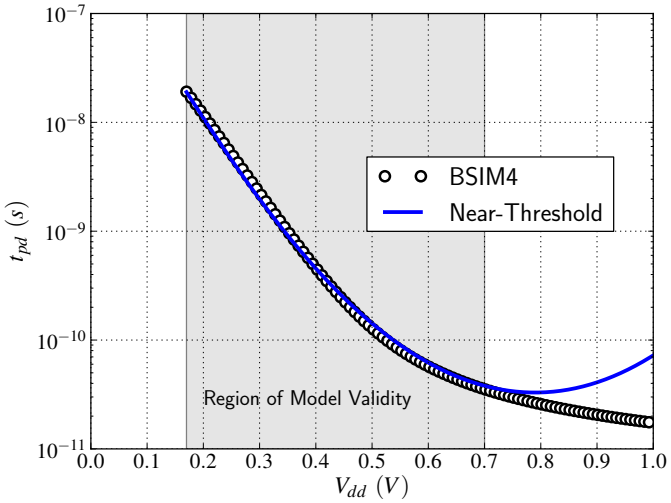


Figure 2.6: Inverter FO4 delay, Equation 2.44, plotted for entire V_{DD} range against a BSIM4 SPICE simulation of 65-nm general-purpose process at 70°C for minimum-size inverter driving an FO4 load. Fit from 135mV to 700mV yielding, $V_t = 386\text{mV}$, $n = 1.43$, and $\frac{C_{\text{load}}}{I_F} = 1.42\frac{ns}{V}$.

Table 2.5: FO4 delay of combinational gates determined using Equation 2.44 and compared to BSIM4 SPICE simulation of 65-nm general-purpose process at 70°C. Fit from 170mV to 750mV with $V_t = 386\text{mV}$ and $n = 1.43$

Gate	$t_{pd}(ns)$ at $V_{DD} = V_t$	$\frac{C_{\text{load}}}{I_F}\left(\frac{ns}{V}\right)$	Maximum Absolute Error	Mean Absolute Error
INV_1X	0.57	1.42	13%	8.3%
INV_4X	0.49	1.29	11%	5.7%
INV_8X	0.48	1.28	10%	5.6%
NAND2	1.2	3.03	19%	11%
NOR2	1.2	3.02	14%	7.8%
AOI21	2.4	5.42	29%	15%

2.3.2 Energy Model

The total energy dissipated by a CMOS circuit, E_{tot}, can be expressed as the summation of a dynamic component, E_{dyn}, corresponding to the charging and discharging of capacitance and a leakage component, E_{leak}, attributed to parasitic leakage current. Assuming periodic operation, *e.g.,* clocking, the total energy can be defined in terms of energy-per-cycle. That is,

$$E_{\text{tot}} = \alpha E_{\text{dyn}} + E_{\text{leak}}, \tag{2.45}$$

where α is the switching activity factor; it models the common case in which only a fraction (alpha) of the logic-gates in the circuit switch. There are numerous physical mechanisms behind leakage currents in modern MOSFETs [76], but in current technology nodes operating in the near-threshold region, sub-threshold drain-to-source current dominates [48]. The leakage energy is thus defined as

$$E_{\text{leak}} = N_l I_{\text{off}} V_{DD} T_c, \tag{2.46}$$

where T_c is the cycle time (typically the critical path delay), and N_l is the number of representative gates that are leaking in a cycle. The dynamic energy of the circuit, E_{dyn}, is defined as

$$E_{\text{dyn}} = C_{\text{dyn}} V_{DD}^2, \tag{2.47}$$

where C_{dyn} represents the entire switching capacitance, *i.e.,* it includes glitching and crowbar current. The cycle time can be defined in terms of a sequence of representative gates and corresponding delays as

$$\begin{aligned} T_c =& t_d, \text{ and} \\ t_d =& t_{pd} L_{dp}, \end{aligned} \tag{2.48}$$

where t_d is the path delay, and L_{dp} is the number of gates on the path each with a delay of t_{pd}.

The off-current, I_{off}, for a single gate can be defined in terms of the sub-threshold drain current from Equation 2.24; letting $V_g = V_s = 0$ and $V_d = V_{DD}$ gives

$$I_{\text{off}} = I_0 e^{\frac{-V_t}{n\phi_t}} \left(1 - e^{\frac{-V_{DD}}{\phi_t}}\right). \tag{2.49}$$

Assuming V_{DD} is a few times larger than the thermal voltage allows the terms in parentheses to be approximated as unity; that is,

$$I_{\text{off}} = I_0 e^{\frac{-V_t}{n\phi_t}}. \tag{2.50}$$

In modern technologies, the inclusion of short-channel effects in the off-current model can significantly improve model accuracy. For example, ignoring the effects of DIBL can result in an order of magnitude of error [42]. The effects of DIBL can be included by explicitly making V_t a function of V_{ds} [91]. That is, the effective threshold voltage becomes $V_t - \eta V_{DD}$, where η is the DIBL factor. Substituting this value into Equation 2.50 (in place of V_t) gives

$$I_{\text{off}} = I_0 e^{\frac{\eta V_{DD} - V_t}{n\phi_t}}. \tag{2.51}$$

Figure 2.7 shows the application of the off-current equation to an NFET and PFET in the 65-nm GP process. The values of fitting bounds, n, and V_t are taken from the t_{pd} model detailed in Figure 2.6. The least-squares fit value for η is 0.134, NFET $I_0 = 6.34\mu A$, and PFET $I_0 = 0.564\mu A$. For the NFET there is a mean

absolute error is 3.4% and a maximum absolute error of 10%; for the PFET there is a mean absolute error is 3.7% and a maximum absolute error of 15%.

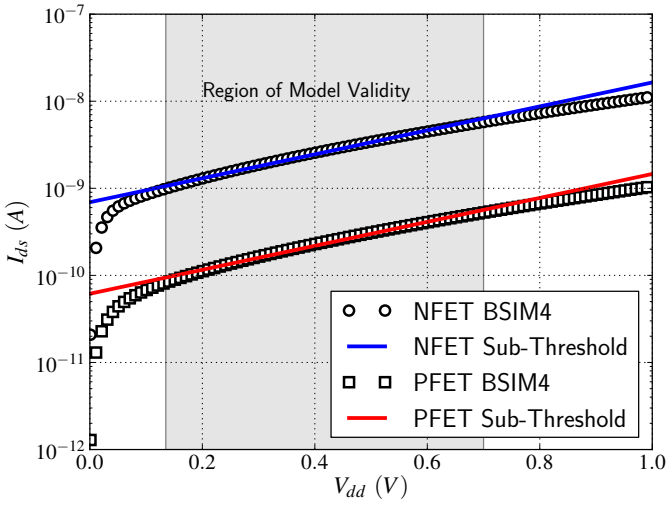


Figure 2.7: Off-current, Equation 2.51, plotted for entire V_{DD} range against a BSIM4 SPICE simulation of 65-nm general-purpose process at 70°C for minimum-size devices with $V_t = 386$mV, $n = 1.43$. Fit from 135mV to 700mV, resulting in $\eta = 0.134$, NFET $I_0 = 6.34\mu A$, and PFET $I_0 = 0.564\mu A$.

The off-current equation (Equation 2.51), can be substituted into the expression for leakage energy (Equation 2.46). That is,

$$E_{\text{leak}} = I_0 V_{DD} N_l T_c e^{\frac{\eta V_{DD} - V_t}{n\phi_t}}. \tag{2.52}$$

Combining this with the dynamic-energy equation (Equation 2.47) by way of Equation 2.45 yields in an expanded expression for energy;

$$E_{\text{tot}} = \alpha C_{\text{dyn}} V_{DD}^2 + I_0 V_{DD} N_l T_c e^{\frac{\eta V_{DD} - V_t}{n\phi_t}}. \tag{2.53}$$

Finally, expanding the term T_c with Equations 2.48 and 2.44 results in the full expression for energy-per-cycle

$$E_{\text{tot}} = \alpha C_{\text{dyn}} V_{DD}^2 + N_l L_{dp} \frac{I_0}{I_F} V_{DD}^2 C_{\text{load}} e^{-k_1 \frac{V_{DT}}{n\phi_t} - k_2 \left(\frac{V_{DT}}{n\phi_t}\right)^2 + \frac{\eta V_{DD} - V_t}{n\phi_t}}. \tag{2.54}$$

Equation 2.54 is continuously differentiable, and can be used to solve traditional and sensitivity-based optimization problems. For example, in the 65-nm GP process, for a chain of FO4 inverters, $C_{\text{dyn}} \approx 1.8fF * L_{dp}$. Using this value for C_{dyn}, with $L_{dp} = N_l = 20$, and the parameters from Figures 2.6 and 2.7, Figure 2.8 gives the minimum-energy operating voltage as a function of activity factor for the 65-nm GP process using Equation 2.54. Figure 2.8 also shows the minimum-energy operating voltage when the weak-inversion approximation (Equation 2.23) and the strong-inversion approximation (Equation 2.27) are used as models for

Table 2.6: Minimum-energy operating voltage error relative to SPICE simulation of BSIM4 model – the corresponding plots are depicted in Figure 2.8

Model	Maximum Absolute Error	Mean Absolute Error
Near-Threshold	5.1%	2.6%
Weak Inversion	34%	11%
Strong Inversion	84%	15%

I_{on}. The errors for these approximations relative to SPICE simulation of the BSIM4 model are listed in Table 2.6. The strong-inversion approximation is a poor model for high activity factors and the weak-inversion approximation is a poor model for low activity factors. The near-threshold model proves to be accurate for a wide range of activity factors.

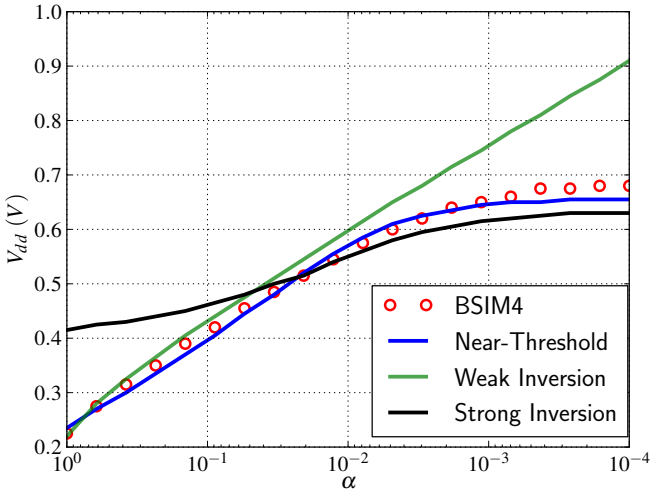


Figure 2.8: Minimum-energy operating voltage vs. activity factor (α). The circuit consists of a linear chain of 20 minimum-size inverters with FO4 loads in a 65-nm general-purpose process at 70°C.

2.3.3 Statistical Delay Model

Timing and delay play a critical role in digital circuit optimization, and in modern technologies the effects of parameter variation on path-delay cannot be ignored. In order to account for parameter variation, static timing analysis (STA)—the most prominent method of delay analysis in digital circuit design—must incorporate statistical methods (*e.g.,* by way statistical static timing analysis (SSTA)) [1, 2, 31]. Parameter variation can be modeled in a number of different ways, and a global corner model with local random variation is accurate, but slightly pessimistic [7]. In this model, global variation affects all devices in the same way (*e.g.,* the TT,

FS, SF, SS corners), and local variation is truly random: *i.e.,* neighboring identically-drawn devices may behave differently. With local variation, the physical effects that dominate parameter variation depend on the operating region. In the sub-threshold and near-threshold regions, parameter variation is dominated by random uncorrelated normally distributed V_t variation [33]. That is, when modeling the delay of circuits operating sub-threshold or near-threshold, for any particular global corner, the effects of parameter variation can be modeled by considering the V_t of each device as an independent normal random variable (RV). The goal of this section is to generate a closed-form stochastic delay model by way of the near-threshold delay model (Equation 2.44) with the new assumption that V_t is an RV.

If X is a normally distributed RV with mean denoted as μ_X, and variance denoted as σ_X^2, then the corresponding probability density function (PDF), $f(X)$, is given by

$$f(X) = \frac{1}{\sigma_X\sqrt{2\pi}} e^{-\frac{(X-\mu_X)^2}{2\sigma_X^2}}. \tag{2.55}$$

For $g(X)$ a function of X, The expected value, E, can be calculated as

$$E[g(X)] = \int_{-\infty}^{\infty} g(X)f(X)dx, \tag{2.56}$$

and the variance, Var, as

$$Var[g(X)] = E[(g(X)^2] - (E[g(X)])^2. \tag{2.57}$$

Treating V_t as a normally distributed RV with mean μ_{V_t} and standard deviation σ_{V_t}, the expected value of t_{pd} can be calculated by applying t_{pd} from Equation 2.44 to Equation 2.56. That is,

$$E[t_{pd}(V_t)] = \int_{-\infty}^{\infty} \frac{C_{\text{load}}}{I_F} \frac{V_{DD}}{\sigma_{V_t}\sqrt{2\pi}} e^{-k_1\frac{V_{DD}-V_t}{n\phi_t} - k_2\left(\frac{V_{DD}-V_t}{n\phi_t}\right)^2 - \frac{(V_t-\mu_{V_t})^2}{2\sigma_{V_t}^2}} dV_t. \tag{2.58}$$

Similarly, applying t_{pd} from Equation 2.44 to Equation 2.57 gives the variance as

$$Var[t_{pd}(V_t)] = \int_{-\infty}^{\infty} \frac{C_{\text{load}}^2}{I_F^2} \frac{V_{DD}^2}{\sigma_{V_t}\sqrt{2\pi}} e^{-2k_1\frac{V_{DD}-V_t}{n\phi_t} - 2k_2\left(\frac{V_{DD}-V_t}{n\phi_t}\right)^2 - \frac{(V_t-\mu_{V_t})^2}{2\sigma_{V_t}^2}} dV_t - (E[t_{pd}(V_t)])^2. \tag{2.59}$$

Due to the form of Equation 2.44, $log(t_{pd}(V_t))$ is an RV with a non-central χ^2 distribution, and $t_{pd}(V_t)$ can be approximated as log-normal with expected value and variance given by Equations 2.58 and 2.59, respectively.[10] The sum of log-normal RVs can be approximated as log-normal [9, 10], giving closed-form

[10] X is a log-normal RV iff $log(X)$ is normally distributed.

Table 2.7: Near-Threshold statistical delay model (Equations 2.60 and 2.61) compared to Monte Carlo SPICE simulations of BSIM4 statistical model for 65-nm GP CMOS from 300mV to 700mV at 100mV intervals (at TT-corner, 70°C, and with 10K MC trials per V_{DD} accounting for local parameter variation) – path delays corresponding to chains of 2, 10, and 20 inverters (with F04 loads at each inverter) are considered

Measurement	Path Length (gates)	Maximum Absolute Error	Mean Absolute Error
$E[t_d]$	2	13%	7.8%
$Var[t_d]$	2	32%	18%
$E[t_d]$	10	13%	8.9%
$Var[t_d]$	10	16%	12%
$E[t_d]$	20	20%	12%
$Var[t_d]$	20	17%	16%

equations for the path delay, t_d, of a sequence of gates with L_{dp} gates on the path (from Equation 2.48).

$$E[t_d(V_t; L_{dp})] = \sum_{i \in \{1,2,\ldots,L_{dp}\}} E[t^i_{pd}(V_t)], \tag{2.60}$$

and

$$Var[t_d(V_t; L_{dp})] = \sum_{i \in \{1,2,\ldots,L_{dp}\}} Var[t^i_{pd}(V_t)], \tag{2.61}$$

where t^i_{pd} is the delay of the i-th gate in the path.

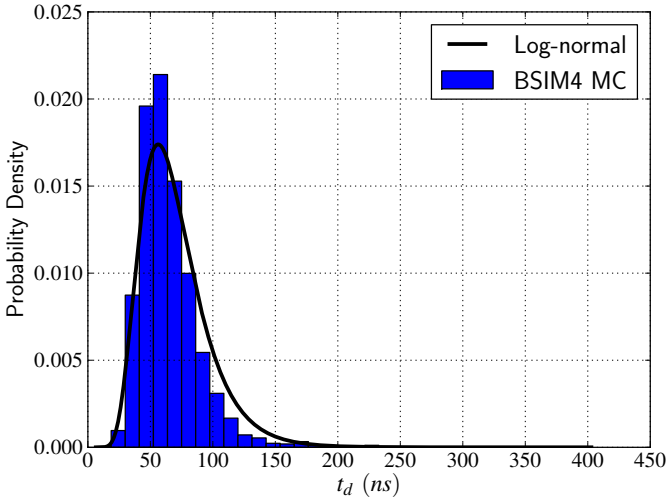


Figure 2.9: Log-normal distribution for path-delay, using an expected value and variance calculated with the near-threshold statistical delay model (Equations 2.60 and 2.61), compared to Monte Carlo SPICE simulations of BSIM4 statistical model for a chain of 20 minimum-size inverters (with FO4 loads at the output of each inverter) in 65-nm GP CMOS with $V_{DD} = 300$mV (at TT-corner, 70°C, and with 10K MC trials accounting for local parameter variation).

With these statistical delay models, short-channel effects cannot be completely ignored. As with the I_{off} model (Equation 2.51), DIBL can be easily incorporated by using an effective threshold voltage of $V_t - \eta V_{DD}$ in lieu of V_t. In the 65-nm GP process, incorporating the effects of DIBL into Equation 2.44 yields new parameters: $\eta = 0.134, n = 1.61, \frac{C_{\text{load}}}{I_F} = 1.23\frac{ns}{V}$. V_t is normally distributed with mean $\mu_{V_t} = 449\text{mV}$ and standard deviation, $\sigma_{V_t} = 56.9\text{mV}$ (computed at the TT-corner from statistical BSIM4 models using the methods from [33]). In order to measure model accuracy, Equations 2.60 and 2.61 are compared to Monte Carlo (MC) simulations using SPICE and foundry provided statistical BSIM4 models. Path lengths of 2, 10, and 20 inverters are considered from 300mV to 700mV (at 100mV intervals) with 10K MC trials at each V_{DD}. The error in both the expected value and the standard deviation are reported in Table 2.7. Figures 2.9 and 2.10 depict the histograms generated from 10K MC trials at 300mV and 700mV respectively with a path length of 20 inverters; the corresponding log-normal distributions with expected value and variance calculated from Equations 2.60 and 2.61, respectively, overlay each histogram.

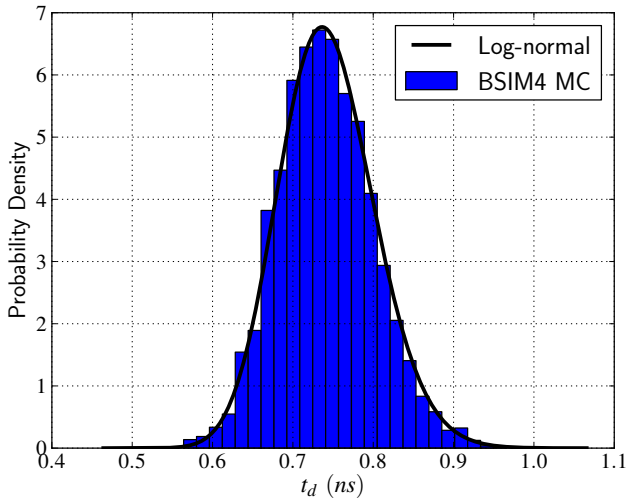


Figure 2.10: Log-normal distribution for path-delay, using an expected value and variance calculated with the near-threshold statistical delay model (Equations 2.60 and 2.61) compared to Monte Carlo SPICE simulations of BSIM4 statistical model for a chain of 20 minimum-size inverters (with FO4 loads at the output of each inverter) in 65-nm GP CMOS with $V_{DD} = 700\text{mV}$ (at TT-corner, 70°C, and with 10K MC trials accounting for local parameter variation).

Approximately 1.3 core-hours of computation time is needed to perform each set of 10K Monte Carlo BSIM4 transient simulations on modern hardware with modern commercial SPICE software. In practice, fewer trials per set may be necessary; however, the computation cost of a broad analysis (*e.g.,* of a large gate set over a wide range of supply voltages at multiple temperatures and multiple global process corners) is significant. The computation cost associated with solving Equations 2.60 and 2.61 is comparatively negligible; the only significant computation expense is incurred when calculating the fitting constants in Equation 2.44: $1.1e-2$ core-hours when V_{DD} is swept from 60mV to $1V$ with a 10mV step size.

2.4 Related Work

MOS modeling dates back many decades, so the set of works that present and discuss various approaches to it are numerous (see [8] or [28] for a historical discussion). The models used in this chapter are based on existing inversion-charge models. The EKV [35, 36, 37] model and the ACM [40] model are examples of accurate and mature continuous compact inversion-charge models. The forthcoming BSIM6 [23] model is a new and purportedly extremely accurate inversion-charge model that is still under development. The work in this chapter differs in that the models are reduced and simplified to the point of limiting applicability to that of digital circuit modeling.

The Lambert W function is currently supported in numerous mathematical computation frameworks, *e.g.,* Maple, Matlab, Mathematica, SciPy. Calhoun used it to define a closed form approximation for the minimum energy operating point of CMOS circuits in [21], and Ortiz-Conde used it to model diode current [71] and surface potential in an undoped-body MOSFET [70].

One of the primary goals of this chapter is to give a simple, continuous digital MOSFET model that can be used for hand-calculations of circuits operating near-threshold. A number of works, *e.g.,* [21] and [97], perform digital circuit analysis in this region ([99] includes variability and derives a sophisticated sub-threshold statistical delay model), but these works rely on the weak-inversion approximation. The weak-inversion model is inaccurate at and above the device threshold voltage, which makes it difficult to perform analysis or establish trends for circuits operating near-threshold. The authors of [38] and [57] address this shortcoming by using the EKV approximation, but this makes hand analysis nearly impossible. Simple, continuous, but piecewise empirical models such as [68] exist but have inaccuracies around the threshold voltage. The initial work for this chapter, presented in [42], is the first publication to present a simplified continuous transregional model which is accurate near-threshold; however, the model presented in [42] is purely empirical, so it lacks the rigor and fitting constant stability associated with the analytical model presented in this chapter.

2.5 Conclusion

This chapter presents the near-threshold model (Equation 2.41), a simplified transregional MOS drain-current model designed specifically for digital circuit analysis of near-threshold circuits. The near-threshold model is continuous and continuous in the first derivative, and it accurately models I_{on} over a wide supply voltage range. The model derivation follows that of previous inversion-charge based models with the addition of a new exact expression for inversion charge (Equation 2.19) and a new simplified inversion-charge approximation (Equations 2.34, 2.35, 2.36). The exact expression for inversion charge may improve the accuracy of certain analyses, *e.g.,* small-signal; verifying this is left as future work. The near-threshold model is validated in four modern CMOS technologies against BSIM4 SPICE simulations, and it is used to solve a circuit analysis problem: finding the minimum-energy operating point of a digital circuit.

As with all models, the near-threshold model has limitations. In a technology with extremely high or low nominal threshold voltages, the model accuracy may degrade. In the technologies examined in this chapter, the HVT devices tended to have higher error rates than the regular devices (see Table 2.4). Explicitly including short-channel effects within the model may somewhat mitigate this problem; however, this is left as future work. Similarly, model accuracy may degrade when modeling I_{on} as a function of transistor length or width unless short-channel effects are explicitly included (as discussed in Section 2.2.3). This problem is apparent in most compact models, but examining it in the context of the near-threshold model is left as future work.

Chapter 3

Quantifying Near-Threshold CMOS Circuit Robustness

3.1 Introduction

It is difficult to design efficient and robust modern digital systems; the sheer complexity of utilizing upwards of a billion devices [16] necessitates the use of numerous levels of logical abstraction throughout the design flow. Errors introduced at different levels of abstraction can result in circuits that fail to function as expected for a number of reasons (*e.g.,* timing, design, and functional failures) [96]. Understanding and quantifying these different modes of failure is important, but failures in the *base digital assumption* supersede all other failures. If a gate cannot switch between logic values, then it cannot perform computation, and assuring correctness with respect to *e.g.,* timing, is moot. Functional failures of this sort can be further divided into many classes [44]; the focus of this chapter is on active device parametric failures [65], *i.e.,* failures caused by one of the most significant hurdles for the future of CMOS scaling [48]: parameter variation.

Parameter variation is caused by stochastic process variation and intrinsic parameter fluctuations (IPF); it is the primary reason why modern digital circuits that function at the process nominal supply voltage (V_{DD}) eventually fail as the supply is lowered [4]. More importantly, parameter variation makes functional digital circuits less robust and hence less reliable [3, 4, 15, 21, 24, 25, 41, 54, 95]. This reduction in robustness may be of little consequence at the process nominal V_{DD}, but, as V_{DD} is lowered, it becomes a critical design concern. Problematically, in order to minimize the power consumption and energy demands of modern digital CMOS circuits, the supply voltage must be scaled sub-threshold or near-threshold [21, 22, 34, 42, 52, 57, 96, 97]. As such, in order to build reliable low-power digital systems, it is essential to quantify circuit robustness as a function of parameter variation, which is the primary goal of this chapter.

The prevailing trend is to perform a simple statistical analysis of worst-case gates and to choose a minimum V_{DD} above which most (or many) gates are likely to function despite parameter variation [4]. The problem with this type of analysis is that it may not be sufficient in real circuits due to the presence of electrical noise. Noise can be mitigated but is fundamentally unavoidable and has proven to be a limiting effect

in engineering digital systems for decades [83]. This chapter proposes a metric and method with which to quantify circuit robustness in terms of parameter variation with respect to noise. Moreover, the method presented is efficient and scalable. The computationally expensive component is limited to a small set of cells that make up modern standard cell libraries and memories, and the calculation of robustness cost is linear in the number of instances of these cells (typically in the range of millions to billions).

The remainder of this chapter is organized as follows. Section 3.2 reviews background material on parameter variation and circuit noise analysis. Section 3.3 introduces the notion of circuit robustness and static noise margins. Section 3.4 details the method for calculating robustness for inverters, and Section 3.5 extends the method to a larger set of CMOS gates. Section 3.6 discusses related works, and finally, Section 3.7 concludes the chapter and discusses potential future research.

3.2 Background

3.2.1 Parameter Variation

In modern CMOS technologies, device parameters such as channel length, oxide thickness, dopant concentration, etc. can have significant deviations from their nominal values due to process-induced and intrinsic parameter fluctuations [12]. Process variability can be considered a global, predictable, and gradual skew in device characteristics introduced by the complexity of manufacturing chips [7] (*e.g.,* from thermal gradients during fabrication [69]). Intrinsic parameter fluctuations are truly statistical in nature and cause significant deviations from device to device within a chip. Intrinsic variations can be attributed to atomistic effects (*e.g.,* random dopant fluctuation (RDF)) and device structure variations (*e.g.,* line edge roughness (LER)) [7, 12, 26]. There are a number of different ways to characterize and partition these effects, and the approach used in this chapter is to consider a global component wherein all devices on a chip are affected in the same way, and a local component wherein each device on a chip has a number of statistical parameters drawn from distributions with mean values set by the global skew. This style of partitioning variation is not as accurate as a full combined statistical model, but it is a good, albeit slightly pessimistic approximation [7].

Considering variation in terms of a global and a local component simplifies statistical analysis and still permits the circuit designer to choose, for example, a worst-case 3σ global corner wherein the die that fall outside of this range are assumed not to yield and should not be optimized for. For circuits operating sub-threshold, the local component of variation is dominated by RDF and is accurately modeled by normally distributed uncorrelated device threshold (V_t) variation [33]. Near-threshold, local variation does exhibit some degree of spatial correlation, and at the process-nominal V_{DD} spatial correlation is significant and cannot be ignored. This increase in the spatial correlation of local variation as a function of V_{DD} can be attributed to the fact that channel-length variation has little effect on devices operating sub-threshold but becomes the dominant effect at approximately twice the threshold voltage [33]. Channel length variation is spatially cor-

related between devices within some radius, and is straightforward to model [7, 33, 39]. Given that the focus of this chapter is to quantify the robustness of low-power sub-threshold and near-threshold circuits, local parameter variation is treated as random and uncorrelated; however, the effects of spatial correlation can be included. Furthermore, SPICE simulations, along with foundry-provided statistically-extracted BSIM4 models, are used throughout this chapter as a basis for correctness; these models are considered accurate over the entire device operating range [6].

3.2.2 Circuit Noise

Circuit noise can be partitioned into a physical component (*e.g.,* thermal noise) and a man-made digital switching component [83]. The dominant sources of physical noise in modern CMOS (which have significant impact on RF CMOS circuits) are $1/f$ noise and thermal noise [80]. Switching noise is caused by the rapid full-rail voltage swings typical in digital systems, and includes cross-talk (due to capacitive and inductive coupling), charge sharing, supply-rail and ground noise, and substrate noise. These switching-noise sources dominate physical noise by several orders of magnitude in digital circuits, and they must be accounted for in the design margins in order to build robust digital systems (even in the absence of appreciable parameter variation) [84]. Accurate modeling of each switching-noise source is possible, but highly impractical for the simulation and analysis of large circuits (millions or billions of devices). It is, however, possible to lump all switching-noise sources together into equivalent series voltage sources between gates [84]. These noise voltage sources are most accurately modeled as time-varying (*i.e.,* AC) sources [32], but using a static DC voltage is an acceptable approximation [83].

3.2.3 Static DC Analysis

Logic gates in modern technologies exhibit a number of frequency-dependent effects, and incorporating these effects greatly increases the complexity of analysis. Fortunately, static DC analysis has proven to be an excellent basis for a wide range of digital circuit characterizations. The first works to discuss the requirements for functional digital circuits [46, 47, 55] exclusively perform DC analysis. Numerous modern works, *e.g.,* [3, 20, 63], also rely on the DC analysis of digital circuits, because in the context of determining functionality, noise resilience, and reliability, it is representative. Moreover, as discussed in Section 3.1, timing failures (which cannot be quantified with DC analysis alone) fall outside of the scope of this work. In this chapter static DC conditions are assumed throughout, and the corresponding canonical method of analysis, voltage transfer characteristics (VTCs)—the static output voltage of a gate as a function of input voltage—are used extensively.

3.3 Defining Circuit Robustness

Parameter variation and noise have a significant impact on circuit robustness, and the primary goal of this chapter is to quantify this impact. To that end, it is necessary to define the notion of robustness with the intuition that increasing parameter variation tends to reduce robustness to noise. Consider two circuits, C_1 and C_2, operating at the same supply voltage; C_1 is more robust than C_2 if and only if C_1 can tolerate more noise than C_2. That is, as the circuit noise increases, C_2 fails to function before C_1. With statistical parameter variation, the notion of failure naturally becomes a probability. Robustness can be defined such that C_1 is more robust than C_2 if and only if for the same quantity of noise in both circuits the probability that C_1 fails is less than the probability that C_2 fails.

As discussed in Section 3.1, the failures of interest are active device parametric failures, wherein a gate or memory erroneously changes state (between binary digital values) because of parameter variation. Circuit noise acts to make these failures more likely, and robust circuits need to function correctly despite parameter variation and switching noise. In order to quantify functional failures due to variation and noise it is necessary to define what it means for a gate or memory to change state. Toward this, consider the *base digital assumption*: the abstraction of networks of transistors as logic gates, and logic gates as Boolean functions over Boolean logic values. This abstraction relies on the definition of a mapping between logic-values and a physical quantity: the electrical potential of charge stored on capacitive gate nodes. In the simplest mapping, nodes near the supply rail potential, V_{DD}, represent a logic-`1`, and nodes near GND represent a logic-`0`; however, it is surprisingly difficult to define *near*. That is, it is difficult to give an exact (necessary and sufficient) mapping between node voltages and logic values for an arbitrary network of logic-gates, because each logic-gate *interprets* input voltages differently.

In a real CMOS circuit, no two gates are identical. They differ in function, topology, and sizing; and distinct instances of the same gate differ because of parameter variation. Consider an inverter; if a `0` is applied to its input, then a `1` is produced on its output. Similarly, a `1` at the input results in a `0` at the output. The problem is that it is possible—by way of intentional construction or parameter variation—to have two distinct inverters, INV_1 and INV_2, that behave differently. Suppose that for input voltages near V_{DD} or GND, INV_1 and INV_2 behave logically identically and correctly (*i.e.,* they invert), but for some input voltage, V_X, between V_{DD} and GND, INV_1 produces a `0` on its output and INV_2 produces a `1`. In this situation, INV_1 and INV_2 *interpret* V_X differently. The situation is further complicated when the notion of the output voltage level is considered. That is, the output of INV_1 is really only a `0` when a subsequent gate *interprets* it as such, and so on down a chain of gates.

Since different gates have different *interpretations* of input voltages, the exact mapping between voltage levels and logic values needs to be defined in terms of this *interpretation* (as opposed to using a global bound). That is, suppose that worst-case boundaries on voltages are defined by V_H and V_L, where it is known that all gates in a circuit *interpret* voltages above V_H as a `1` and all voltages below V_L as `0`; then the mapping

of $V(G) > V_H \leftrightarrow \texttt{1}$ and $V(G) < V_L \leftrightarrow \texttt{0}$ is sufficient for some notion of correct operation, but it is not necessary. This distinction is important, because this sort of worst-case definition is simple but not practical for the analysis of modern low-voltage circuits.

Consider an example that demonstrates the trouble with using the worst-case definitions for V_H and V_L in low-voltage applications. Figure 3.1 depicts the VTCs for 100 instances of a minimum-size inverter in a modern 40-nm low-power bulk CMOS process with $V_{DD} = 200\text{mV}$; the curves vary significantly due to random parameter variation. These VTCs have remarkably similar shapes and are nearly identical modulo horizontal translation. As such, it is reasonable to consider defining $V_H = 180\text{mV}$ and $V_L = 20\text{mV}$ as worst-case output high and low voltages, respectively (these boundaries are also depicted by blue and red lines respectively in Figure 3.1). The problem with this worst-case output mapping is that the corresponding input voltages that yield a logical-1 on the output then range from 25mV to 150mV; similarly, the input voltages that yield a logical-0 on the output range from 65mV to 195mV. These ranges overlap, so a worst-case mapping of input voltages to logic values cannot be defined (the nonsensical worst-case mapping would be $V(G) > 65\text{mV} \leftrightarrow \texttt{1}$ and $V(G) < 150\text{mV} \leftrightarrow \texttt{0}$).

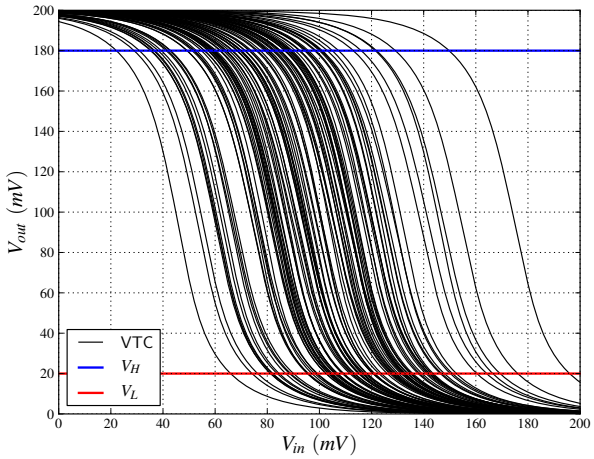


Figure 3.1: Voltage transfer characteristics for 100 Monte Carlo trials of a minimum-size inverter in a commercial 40-nm low-power CMOS process utilizing foundry provided statistical models for local random parameter variation at the TT global corner ($V_{DD} = 200\text{mV}$ at $25°C$ TT-Corner).

3.3.1 Static Noise Margin

A better approach to defining a local notion of *interpretation* stems from static noise margin (SNM) analysis. The static noise margin of cross-coupled inverters was first presented in [46, 47] and later clarified in [43] and [56]. Consider Figure 3.2; the SNM of this cross-coupled pair represents the largest DC noise voltage, V_{noise}, that can be applied between the bistable pair before the inverters switch state (between logic-0 and logic-1). If the SNM of a cross-coupled pair is less than or equal to zero (*e.g.,* due to parametric variation), then the

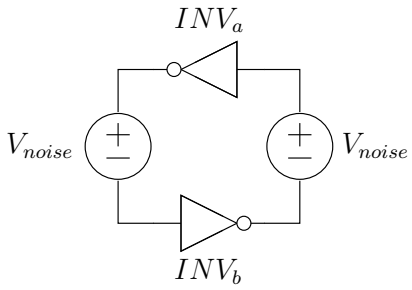


Figure 3.2: Cross-coupled inverter pair and DC noise voltage sources.

pair is not bistable; *i.e.,* it is unable to hold two distinct logic states (a functional failure). If the SNM of the pair is infinitesimally greater than zero, then the cell can hold two distinct logic states, but a diminutive noise can act to switch these states, so the cell is not robust. Given that noise is always present, all cross-coupled pairs of inverters in a digital system must have static noise margins in excess of the system noise in order to maintain state.[1]

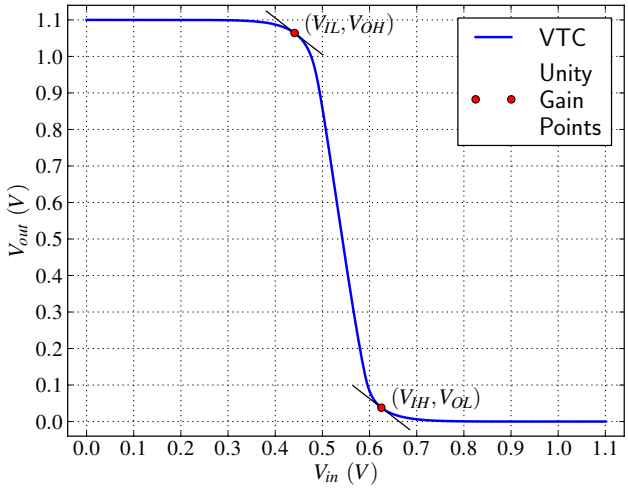


Figure 3.3: Voltage transfer characteristic for a minimum-size inverter in a commercial 40-nm low-power CMOS process ($V_{DD} = 1.1V$ at $25°C$). The unity gain points are used to define the VTC parameters: $V_{OH}, V_{OL}, V_{IH}, V_{IL}$.

There are several mathematically equivalent methods used to measure static noise margins [56]. One such method involves analyzing the unity gain points ($|\frac{dV_{out}}{dV_{in}}| = 1$) of the voltage transfer characteristic. Consider INV_a (INV_b) from Figure 3.2: a static CMOS inverter consisting of a single NFET and PFET, with the VTC depicted in Figure 3.3. Both the functionality of the inverter and the definition of SNM rely on two properties of the VTC holding: (1) two unity gain points exist and (2) the slope between the unity gain

[1] In real memories, *e.g.,* SRAM arrays, the SNM during both reading and writing of cells need to be considered [20]. Furthermore, ensuring a SNM of greater than zero is necessary, but it may not be sufficient for ensuring read stability and write-ability [41].

points exceeds unity in absolute value [63]. From these unity gain points, four properties of an inverter VTC can be defined: $V_{OH}, V_{OL}, V_{IH}, V_{IL}$, as in Figure 3.3 (see [43] for details). (These four points are referred to as VTC parameters throughout.) The VTC parameters serve to demark definable boundaries between the voltages that are *interpreted* as a logic-`1` or logic-`0`, and the undefined region of high-gain in between. That is, V_{IH} can be considered the lowest voltage that the inverter correctly *interprets* as a `1`, and V_{IL} as the highest voltage that it correctly *interprets* as a 0. Similarly, V_{OH} can be considered the lowest voltage that the inverter will output as a 1 and V_{OL} the highest voltage that the inverter will output as a `0`.

In general, when one gate *drives* another gate, a static noise margin can be defined. This static noise margin can be broken into two components: a noise margin high (NM_H) and a noise margin low (NM_L) (one for each logic value). Consider a pair of inverters, with INV_x driving INV_y. The two components of the corresponding noise margin are defined as

$$NM_H(INV_x, INV_y) = V_{OH}(INV_x) - V_{IH}(INV_y), \tag{3.1}$$

and,

$$NM_L(INV_x, INV_y) = V_{IL}(INV_y) - V_{OL}(INV_x). \tag{3.2}$$

The static noise margin is defined as the smaller of NM_H or NM_L.

$$\begin{aligned} &SNM(INV_x, INV_y) = \\ &min\left(NM_L(INV_x, INV_y), NM_H(INV_x, INV_y)\right). \end{aligned} \tag{3.3}$$

These relations are implicit functions of V_{DD}.[2]

For cross-coupled inverters, as in Figure 3.2, INV_a drives INV_b, and INV_b drives INV_a, so two different static noise margins can be defined, $SNM(INV_a, INV_b)$ and $SNM(INV_b, INV_a)$. With a few assumption about the VTCs,[3] the condition that $SNM(INV_a, INV_b) > V_{noise} \cap SNM(INV_b, INV_a) > V_{noise}$ is a necessary and sufficient condition for differentiation of binary logic-values by way of the electrical potential stored on the output of each inverter [46, 47, 56]. The static noise margin of cross-coupled inverters plays an important role in quantifying circuit robustness, but the notion must be extended to incorporate parametric variability and generalized in order to apply it to arbitrary networks of gates.

3.3.2 Statistical Robustness

This section defines a robustness metric for cross-coupled inverters that includes parameter variation and noise by way of a statistical noise margin constraint. When considering two different circuits, C_1 and C_2, operating with the same supply voltage, C_1 is more robust than C_2 if and only if for the same quantity of

[2] Equations 3.1, 3.2, and 3.3 (and all dependent equations) are actually implicit functions of all operating parameters, *e.g.,* temperature, body potentials, etc.

[3] The VTCs must be monotonic and have a single inflection point.

noise in both circuits the probability that C_1 fails is less than the probability that C_2 fails. That is, for two different circuits C_1 and C_2,

$$ROB(C_1) > ROB(C_2) \leftrightarrow P(FAIL(C_1)) < P(FAIL(C_2)), \tag{3.4}$$

where ROB corresponds to circuit robustness and $FAIL$ to circuit failure.

Switching noise in digital circuits can be estimated with known-methods [83, 84], and, as with other common metrics, *e.g.,* power and cycle time, it can be reduced and optimized for (typically at some cost; *e.g.,* spreading wires reduces coupling noise at the expense of area). As such, the circuit designer can choose a noise margin target, NM_T: a minimum noise margin constraint for all gates.[4] If any gate has a noise margin less than or equal to the NM_T, then the gate is said to fail, as is the entire circuit containing the failing gate. Consider a cross-coupled inverter-pair, INV_a and INV_b, (as in Figure 3.2 with $V_{noise} = 0V$) operating at a particular V_{DD}. The probability of failure for a pair can then be defined such that

$$\begin{aligned} & P(FAIL(INV_a, NM_T) \cup FAIL(INV_b, NM_T)) \\ = & P(SNM(INV_a, INV_b) \leq NM_T \cup \\ & SNM(INV_b, INV_a) \leq NM_T). \end{aligned} \tag{3.5}$$

For a circuit, C_a, consisting of n cross-coupled inverter-pairs, *i.e.,* $C_a = (INV_a^i, INV_b^i)$ for $i \in \{1, 2, ..., n\}$,

$$\begin{aligned} & P(FAIL(C_a, NM_T)) = \\ & P\left(\bigcup_{i \in \{1,2,...,n\}} FAIL(INV_a^i, NM_T) \cup FAIL(INV_b^i, NM_T) \right). \end{aligned} \tag{3.6}$$

These two relations treat both the probability of failure and SNM as random variables (RVs). In order to compute these quantities, the corresponding distributions and the effects of correlation are considered in Section 3.4. These two relations are generalized for application to arbitrary networks of gates in Section 3.5.

3.4 Calculating Robustness

One of the goals of this chapter is to define a method for calculating robustness in such a way that it can be feasibly computed for large circuits (billions of gates), and which also fits in with the most prevalent method of system design, *i.e.,* standard-cell hierarchical digital circuit design. This necessitates the construction of a new compact model for statistical robustness with parameters that can be stored alongside timing and energy data in standard cell libraries. Moreover, the model must be defined such that the compact data

[4] A unique noise margin target can be chosen for each gate (if desired). In this way, *noisy* gates can be assigned larger targets than *quiet* gates.

is composable; *i.e.,* the robustness of an arbitrary network of standard cells must be computable by the composition of robustness data from member cells. In this way, the robustness of a large circuit (built out of standard cells) can be readily calculated.

3.4.1 Statistical VTC Parameters

Device parameter variation results in variation in the static noise margins of gates; the precise relationship depends on the type of parameter variation and the device operating regime (sub-threshold see [3, 20], and above threshold see [13, 81]). The variation in SNM can be analyzed in terms of NM_H and NM_L variation (see Equation 3.3). Similarly, NM_H and NM_L can be considered in terms of the corresponding constituent VTC parameters, V_{OH}, V_{IH}, and V_{OL}, V_{IL}, respectively (see Equations 3.1 and 3.2). In modern bulk CMOS technologies, the output VTC parameters of a gate, V_{OH} and V_{OL}, can be considered regular (not random) variables.[5] The input VTC parameters, V_{IH} and V_{IL}, are normal random variables [20]. Consider Figure 3.1 (in Section 3.3): for a particular gate (an inverter) operating at a particular supply voltage (200mV) the output VTC parameters, V_{OH} and V_{OL}, are nearly constant and close to V_{DD} and GND, respectively (consider the blue and red lines). The horizontal translation between this family of VTC curves—due to random parameter variation—corresponds to shifts in the input VTC parameters, V_{IH} and V_{IL}.

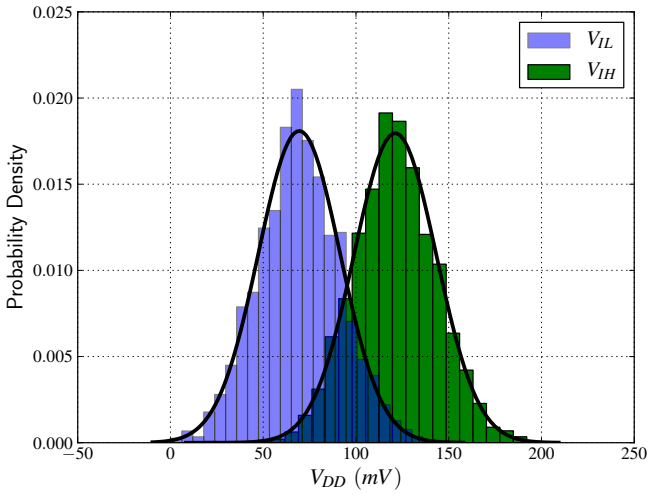


Figure 3.4: V_{IH} and V_{IL} distributions for a minimum-size inverter in a commercial 40-nm low-power CMOS process at the TT-Corner ($V_{DD} = 200$mV at $25°C$).

The input VTC parameter are normally distributed with mean and standard deviation determined by the supply voltage, gate topology, temperature, and global corner. This is confirmed by the analysis of two standard cell libraries in different technologies and from different foundries (a 40-nm low-power process and

[5] First-order analysis in [3] finds V_{OH} and V_{OL} to be global constants dependent only on temperature when operating in the sub-threshold regime. Including second order affects and near-threshold operation induces a dependence on V_{DD} and gate topology, so V_{OH} and V_{OL} are treated as regular variables.

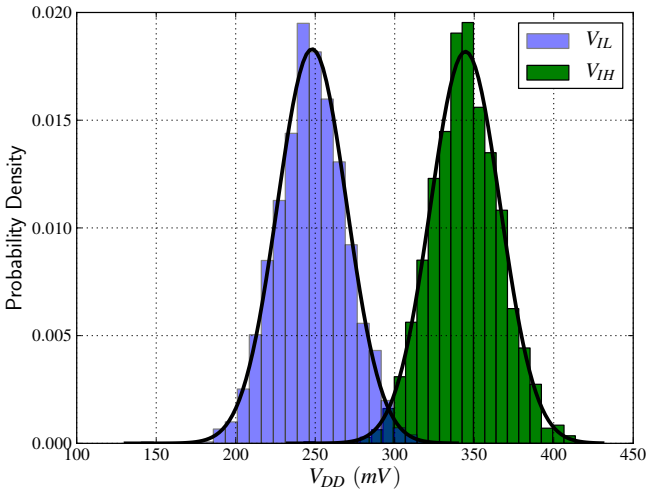


Figure 3.5: V_{IH} and V_{IL} distributions for a minimum-size inverter in a commercial 40-nm low-power CMOS process at the TT-Corner ($V_{DD} = 600$mV at $25°C$).

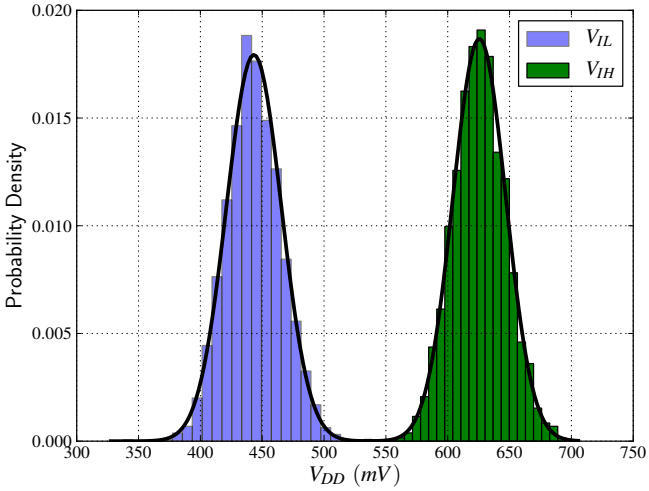


Figure 3.6: V_{IH} and V_{IL} distributions for a minimum-size inverter in a commercial 40-nm low-power CMOS process at the TT-Corner ($V_{DD} = 1.1V$ at $25°C$).

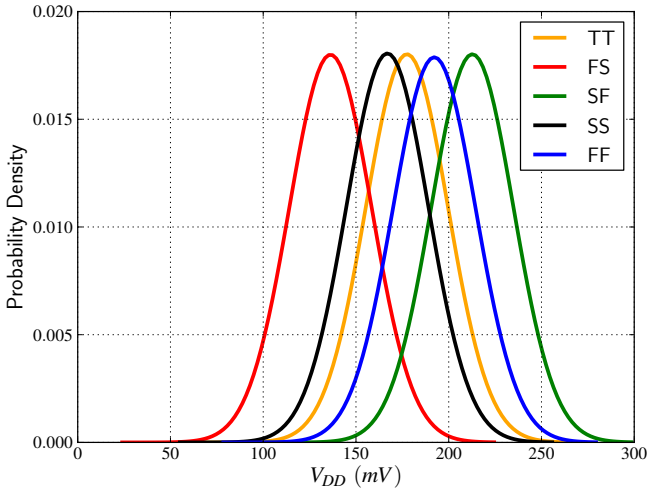


Figure 3.7: V_{IH} distributions for a minimum-size inverter in a commercial 40-nm low-power CMOS process ($V_{DD} = 300$mV at $25°C$). Global variation shifts the mean value for both V_{IH} and V_{IL}.

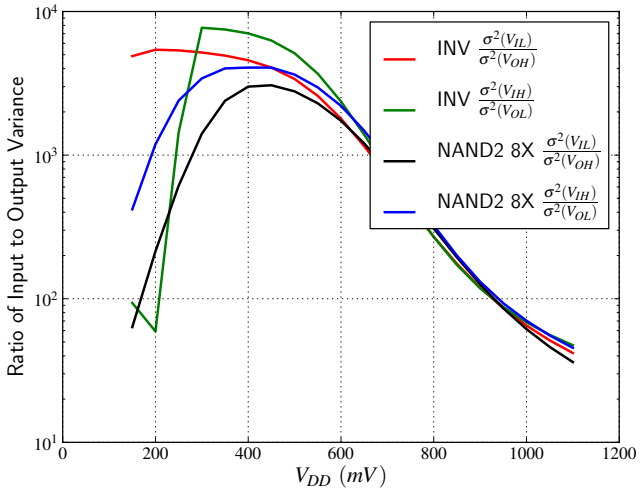


Figure 3.8: Ratio of input VTC parameter variance to output VTC parameter variance in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner). The large ratio across the entire operating range makes it possible to approximate the output VTC parameters as regular variables, whereas the input VTC parameters are considered random variables.

a 65-nm low-power process). Both cell libraries contain hundreds of cells, and Anderson-Darling normality testing shows that neither V_{IH} nor V_{IL} have any significant departure from normality over the entire operating range.[6] Figures 3.4, 3.5, and 3.6 depict V_{IH} and V_{IL} histograms along with corresponding normal probability density functions (PDFs) for a minimum-size inverter operating sub-threshold, near-threshold, and at process nominal V_{DD}, respectively. Global variation simply skews the mean value, as depicted in Figure 3.7. Finally, Figure 3.8 further justifies the treatment of the output VTC parameters as regular variables: the spread of each input VTC parameter is several orders of magnitude greater than the corresponding output VTC parameter spread.

3.4.2 Statistical Noise Margins

At any particular global corner, local parameter variation is uncorrelated (see Section 3.2.1), so the VTC parameters for distinct gates are independent. Consider two distinct inverters, INV_x driving INV_y; INV_x and INV_y have independent normally distributed input VTC parameters. From Equations 3.1 and 3.2 *and* the assumption that the corresponding output VTC parameters are regular variables, it follows that the corresponding NM_H and NM_L are also normally distributed RVs with mean and standard deviation given by

$$\begin{aligned} \mu(NM_H(INV_x, INV_y)) &= V_{OH}(INV_x) - \mu(V_{IH}(INV_y)), \\ \sigma(NM_H(INV_x, INV_y)) &= \sigma(V_{IH}(INV_y)), \end{aligned} \tag{3.7}$$

and

$$\begin{aligned} \mu(NM_L(INV_x, INV_y)) &= \mu(V_{IL}(INV_y)) - V_{OL}(INV_x), \\ \sigma(NM_L(INV_x, INV_y)) &= \sigma(V_{IL}(INV_y)), \end{aligned} \tag{3.8}$$

where for any RV Z, $\mu(Z)$ and $\sigma(Z)$ denote the mean value and the standard deviation, respectively. Inconveniently, the statistical SNM does not follow directly from Equation 3.3 (due to the min function). If $NM_H(INV_x, INV_y)$ and $NM_L(INV_x, INV_y)$ are independent, order statistics can be used to directly calculate $SNM(INV_x, INV_y)$ [20]; however, they are not independent. From Figure 3.9, it is clear that the input VTC parameters are highly positively correlated, and it follows from this and Equations 3.7 *and* 3.8 that NM_H and NM_L are highly negatively correlated, which makes the direct calculation of SNM difficult. The approach taken in this dissertation is to use NM_H and NM_L directly to calculate the probability that a circuit fails, thus avoiding the need to compute SNM. In this way, the effects of correlation can be accounted for, and a general method for failure analysis is made possible.

[6] The nature of normality testing makes it difficult to make a stronger statement. Furthermore, it is extremely difficult to verify that the tails of purportedly normal distributions are actually normal; as such, treating V_{IH} and V_{IL} and normal RVs should be considered an approximation.

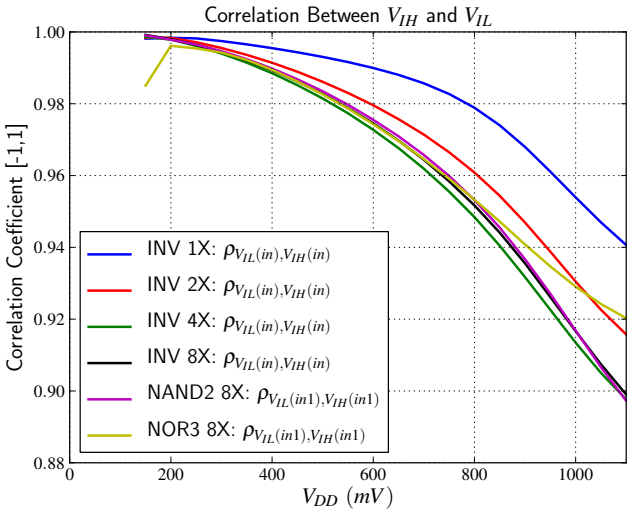


Figure 3.9: Correlation between V_{IH} and V_{IL} in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner). These input VTC parameters are highly positively correlated across V_{DD} for a wide variety of gates.

3.4.3 Cross-coupled Inverter: Failure Probability

With a notion of statistical robustness (Section 3.3.2) and statistical noise margins (Section 3.4.2) defined, it is possible to calculate the probability of cross-coupled inverter failure, and hence its robustness. Again, consider a cross-coupled inverter-pair, INV_a and INV_b, (as in Figure 3.2 with $V_{noise} = 0V$) operating at a particular V_{DD} and with a noise margin target of NM_T. Calculating the probability of failure (from Equation 3.5) necessitates the evaluation of $P(SNM(INV_a, INV_b) \leq NM_T \cup SNM(INV_b, INV_a) \leq NM_T)$. Assuming statistical independence, the disjunction can be treated as an addition, and Equation 3.5 reduces to

$$\begin{aligned} & P(FAIL(INV_a, NM_T) \cup FAIL(INV_b, NM_T)) \\ = \; & P(SNM(INV_a, INV_b) \leq NM_T) + \\ & P(SNM(INV_b, INV_a) \leq NM_T). \end{aligned} \tag{3.9}$$

To calculate this quantity in closed-form, it is necessary to re-term this relation using NM_H and NM_L in lieu of SNM (as discussed in Section 3.4.2). In order to do this, upper and lower bounds on failure are determined, and then an approximation is given.

3.4.3.1 Upper Bound

If $SNM(INV_a, INV_b) \leq NM_T$, then $NM_H(INV_a, INV_b) \leq NM_T$ and/or $NM_L(INV_a, INV_b)$

$\leq NM_T$ (this follows directly from Equation 3.3). This can be stated in terms of probabilities as

$$\begin{aligned} & P(SNM(INV_a, INV_b) \leq NM_T) \\ \leq\ & P(NM_H(INV_a, INV_b) \leq NM_T \cup \\ & NM_L(INV_a, INV_b) \leq NM_T). \end{aligned} \tag{3.10}$$

Due to the high degree of anti-correlation between NM_H and NM_L (see Section 3.4.2), the disjunction can be approximated as an addition and

$$\begin{aligned} & P(SNM(INV_a, INV_b) \leq NM_T) \\ \leq\ & P(NM_H(INV_a, INV_b) \leq NM_T) + \\ & P(NM_L(INV_a, INV_b) \leq NM_T). \end{aligned} \tag{3.11}$$

Due to symmetry, a similar argument holds for $SNM(INV_b, INV_a)$, so combining Equation 3.9 and 3.11 yields an upper bound on the probability of failure for cross-coupled inverters. That is,

$$\begin{aligned} & P(FAIL(INV_a, NM_T) \cup FAIL(INV_b, NM_T)) \\ \leq\ & P(NM_H(INV_a, INV_b) \leq NM_T) + \\ & P(NM_L(INV_a, INV_b) \leq NM_T) + \\ & P(NM_H(INV_b, INV_a) \leq NM_T) + \\ & P(NM_L(INV_b, INV_a) \leq NM_T). \end{aligned} \tag{3.12}$$

3.4.3.2 Lower Bound

If $NM_H(INV_a, INV_b) \leq NM_T$ and $NM_L(INV_a, INV_b) \leq NM_T$, then $SNM(INV_a, INV_b) \leq NM_T$ (this follows directly from Equation 3.3). This can be stated in terms of probabilities as

$$\begin{aligned} & P(SNM(INV_a, INV_b) \leq NM_T) \\ >\ & P(NM_H(INV_a, INV_b) \leq NM_T \cap \\ & NM_L(INV_a, INV_b) \leq NM_T). \end{aligned} \tag{3.13}$$

Due to the high degree of anti-correlation between NM_H and NM_L (see Section 3.4.2), the conditional probability of each event ($NM_H(INV_a, INV_b) \leq NM_T$, and $NM_L(INV_a, INV_b) \leq NM_T$) is less than

the unconditional probability, so

$$\begin{aligned} & P(SNM(INV_a, INV_b) \leq NM_T) \\ > & P(NM_H(INV_a, INV_b) \leq NM_T) * \\ & P(NM_L(INV_a, INV_b) \leq NM_T). \end{aligned} \tag{3.14}$$

Due to symmetry, a similar argument holds for $SNM(INV_b, INV_a)$, so combining Equation 3.9 and 3.14 yields a lower bound on the probability of failure for cross-coupled inverters. That is,

$$\begin{aligned} & P(FAIL(INV_a, NM_T) \cup FAIL(INV_b, NM_T)) \\ > & P(NM_H(INV_a, INV_b) \leq NM_T) * \\ & P(NM_L(INV_a, INV_b) \leq NM_T) + \\ & P(NM_H(INV_b, INV_a) \leq NM_T) * \\ & P(NM_L(INV_b, INV_a) \leq NM_T). \end{aligned} \tag{3.15}$$

3.4.3.3 Heuristic Approximation

One way to approximate the probability of failure comes from the consideration of the cross-coupled pair as a whole. If INV_A is skewed such that it can barely *interpret* a logical-0 and INV_B is skewed such that is can barely *interpret* a logical-1 (or vice versa), then a failure is likely. That is, if $NM_H(INV_a, INV_b) \leq NM_T$ and $NM_L(INV_b, INV_a) \leq NM_T$, or if $NM_H(INV_b, INV_a) \leq NM_T$ and $NM_L(INV_a, INV_b) \leq NM_T$, then it is likely that $SNM(INV_a, INV_b) \leq NM_T$ or $SNM(INV_b, INV_a) \leq NM_T$. Empirically, with a small shift, δ, the lower bound *approximation* (given by Equation 3.15) leads to an accurate heuristic over a wide range of NM_T and V_{DD}. That is,

$$\begin{aligned} & P(FAIL(INV_a, NM_T) \cup FAIL(INV_b, NM_T)) \\ \approx & P(NM_H(INV_a, INV_b) \leq NM_T + \delta) * \\ & P(NM_L(INV_b, INV_a) \leq NM_T + \delta) + \\ & P(NM_H(INV_b, INV_a) \leq NM_T + \delta) * \\ & P(NM_L(INV_a, INV_b) \leq NM_T + \delta). \end{aligned} \tag{3.16}$$

3.4.4 Probability Computation

Finally, the Gauss error function, erf, and the cumulative distribution function (CDF) of the normal distribution can be used to compute the probability of failure. If Z is a normal random variable with mean μ and

standard deviation σ, and c a constant, then

$$P(Z \leq c) = \frac{1}{2}\left(1 + erf(\frac{c-\mu}{\sigma\sqrt{2}})\right). \tag{3.17}$$

Consider an inverter INV_x driving another inverter INV_y, combining Equations 3.7, 3.8, and 3.17 yields

$$\begin{aligned} &P(NM_H(INV_x, INV_y) \leq NM_T) = \\ &\frac{1}{2}\left[1 + erf\left(\frac{NM_T - (V_{OH}(INV_x) - \mu(V_{IH}(INV_y)))}{\sigma(V_{IH}(INV_y))\sqrt{2}}\right)\right] \\ &\text{and} \\ &P(NM_L(INV_x, INV_y) \leq NM_T) = \\ &\frac{1}{2}\left[1 + erf\left(\frac{NM_T - (\mu(V_{IL}(INV_y)) - V_{OL}(INV_x))}{\sigma(V_{IL}(INV_y))\sqrt{2}}\right)\right]. \end{aligned} \tag{3.18}$$

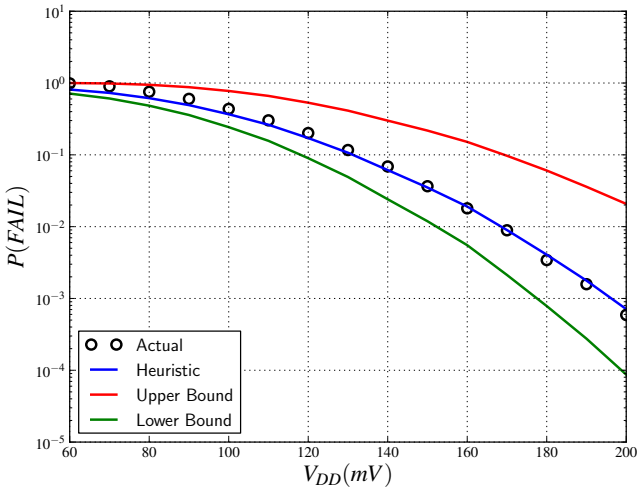


Figure 3.10: Probability of minimum-size cross-coupled inverter-pair failure for $NM_T = 0$mV in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner). For the heuristic approximation, the mean absolute error is 13%, and the maximum absolute error is 20% with $\delta = 4.2\%V_{DD}$.

Equation 3.18 can be applied directly to Equations 3.12, 3.15, and 3.16, thus yielding close-form equations for the probability of cross-coupled inverter failure. Note that the these expressions for failure likelihood rely on an extremely compact set of real numbers:

- $V_{OH}(INV_{x,y})$
- $V_{OL}(INV_{x,y})$
- $\mu(V_{IH}(INV_{x,y}))$
- $\mu(V_{IL}(INV_{x,y}))$

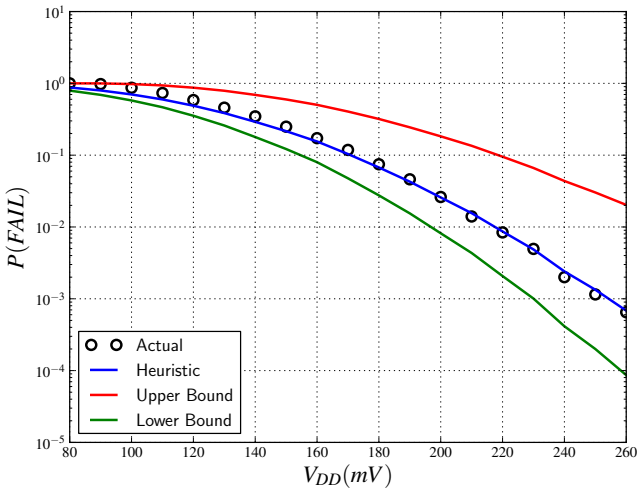


Figure 3.11: Probability of minimum-size cross-coupled inverter-pair failure for $NM_T = 10\%V_{DD}$ in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner). For the heuristic approximation, the mean absolute error is 12%, and the maximum absolute error is 20% with $\delta = 3.2\%V_{DD}$.

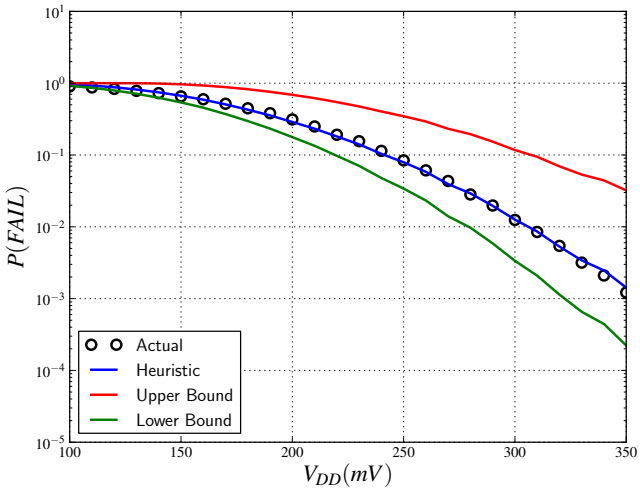


Figure 3.12: Probability of minimum-size cross-coupled inverter-pair failure for $NM_T = 20\%V_{DD}$ in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner). For the heuristic approximation, the mean absolute error is 5.2%, and the maximum absolute error is 17% with $\delta = 2.2\%V_{DD}$.

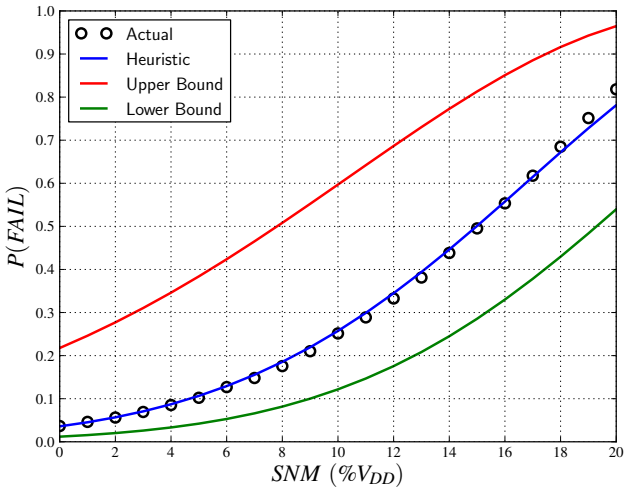


Figure 3.13: Probability of minimum-size cross-coupled inverter-pair failure for $V_{DD} = 150$mV in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner). For the heuristic approximation, the mean absolute error is 2.5%, and the maximum absolute error is 5.5% with $\delta = 4.3\%V_{DD}$.

- $\sigma(V_{IH}(INV_{x,y}))$
- $\sigma(V_{IL}(INV_{x,y}))$,

which is one of the goals of this section. That is, these are the only parameters needed in order to calculate the probability of failure, and hence robustness of a circuit. Figures 3.10, 3.11, and 3.12, plots the probability of failure for a crossed-coupled inverter pair against V_{DD}. Figure 3.13 plot the probability of cross-coupled inverter failure versus NM_T for a fixed supply voltage of 150mV. Digital noise tends to be proportional to V_{DD} [45], so the NM_T is reported as a percentage of V_{DD}. Each of these figures depicts the upper bound, lower bound, and approximation for cross-coupled inverter failure probability, in addition to the actual (empirical) failure rate. Actual failures are calculated via Monte Carlo SPICE simulations with foundry provided statistical BSIM4 models.

Figures 3.10, 3.11, 3.12, and 3.13 also serve to exemplify why an accurate and simple closed-form approximation for the probability of failure is so important. In each of these plots, as V_{DD} increases linearly, the probability of failure decreases exponentially, and the size of the Monte Carlo simulations required to generate accurate failure rates increases exponentially. With a noise margin target of $10\%V_{DD}$ at 300mV the probability of failure is already less than 10^{-5}, so millions of Monte Carlo trials are necessary. A million such trials on modern computers with modern tools requires several core-hours of compute time. Furthermore, it is not uncommon for a modern microprocessor design to contain millions of cross-coupled inverters, so higher supply voltages with lower failure rates on the order of 10^{-9} or lower need to be considered. This corresponds to at least a four order of magnitude increase in compute time for a single temperature and V_{DD} of interest. To ensure reliability, multiple supply voltages and temperatures need to be considered, increasing

the computation requirement by yet another order of magnitude. Optimization of transistor sizing can easily increase the computation requirement by another order of magnitude, resulting in a compute requirement in the realm of millions of core-hours. Finally, one of the goals of this chapter is to extend this type of analysis to arbitrary gates (typical standard cell libraries contain hundreds of cells). This easily pushes the compute requirement to billions of core-hours. A closed-form approximation is more practical.

Finally, the probability of failure of a circuit C_a consisting of n cross-coupled inverter pairs ($C_a = (INV_a^i, INV_b^i)$ for $i \in \{1, 2, ..., n\}$) can easily be computed. Equation 3.6 can be re-written in terms of a global conjunction instead of disjunction as

$$P(FAIL(C_a, NM_T)) = 1 - P\Bigg(\bigcap_{i\in\{1,2,...,n\}} \neg\big(FAIL(INV_a^i, NM_T) \cup FAIL(INV_b^i, NM_T)\big)\Bigg). \tag{3.19}$$

Given the assumption of VTC parameter independence between gate pairs (see Section 3.4.2), the global conjunction can be treated as a product, giving a readily computable compact expression for the probability of failure and hence robustness of a circuit, one of the goals of this section. That is,

$$P(FAIL(C_a, NM_T)) = 1 - P\Bigg(\prod_{i\in\{1,2,...,n\}} \neg\big(FAIL(INV_a^i, NM_T) \cup FAIL(INV_b^i, NM_T)\big)\Bigg). \tag{3.20}$$

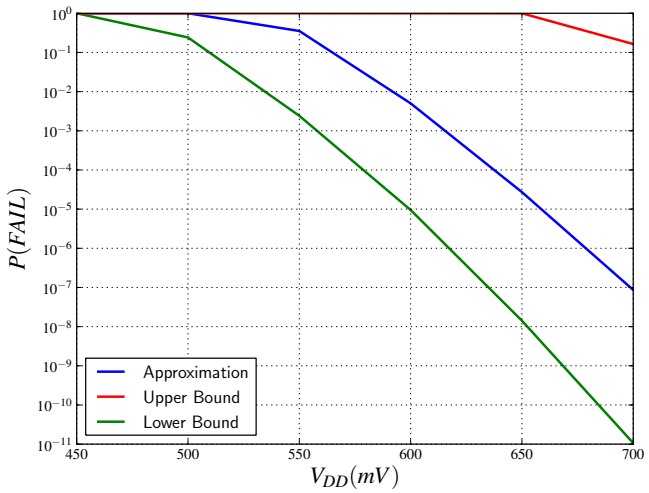


Figure 3.14: Probability of failure for 2e28 minimum-size cross-coupled inverter-pairs with $NM_T = 20\%V_{DD}$ in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner, and $\delta = 2.2\%V_{DD}$).

With Equation 3.20, it is possible to quantify the probability of failure for an entire memory. Figure 3.14 gives the probability of failure for 2e28 independent cross-coupled inverter pairs (*i.e.,* a 32MB mem-

ory). Simulation with statistical SPICE is completely infeasible, so δ is taken from the considerably smaller experiments depicted in Figure 3.12.

3.4.5 Chains of Inverters: Failure Probability

The goal of this section is to extend the notions of noise margins and circuit robustness to arbitrary networks of inverters. This is necessary because the probability of failure of a linear chain of n inverters differs significantly from that of n cross-coupled inverters. At first glance, this seems counter-intuitive; several works (*e.g.,* [56]) have demonstrated that a cross-coupled pair of identical inverters can be modeled as—and is mathematically equivalent to—an infinite chain of identical inverters. Moreover, alternating worst-case (demonic) noise sources between a cross-coupled pair can be modeled as demonic alternating noise in an infinite chain, as depicted in Figure 3.15. The main idea behind this equivalence is that an infinite chain can be viewed as the unrolling of the *loop* that is a cross-coupled pair. When a bistable cross-coupled pair in steady state is perturbed by some voltage δV, the bistable pair either changes digital state, or the inverters act as a restorative filter, successively removing the δV disturbance one *iteration* at a time in the same exact way that a chain of inverters filters a δV disturbance. When the inverters are not identical, the two circuits no longer behave in the same way, and equivalence is lost. The intuition behind why they differ comes from further analysis of the heuristic approximation presented in Section 3.4.3.

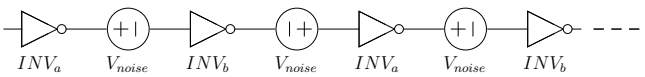


Figure 3.15: Infinite chain construct: equivalent to the cross-coupled pair depicted in Figure 3.2.

Consider a cross-coupled inverter pair (INV_a, INV_b), where INV_a and INV_b behave differently due to parameter variation. With the infinite chain construct, this pair can be modeled as a never-ending alternating linear chain of INV_a driving INV_b driving INV_a driving INV_b, etc. (see Figure 3.15). Suppose that this inverter pair is not robust, *i.e.,* the static noise margin is just slightly larger than 0mV due to INV_a being skewed such that it can barely *interpret* a logical-`0` and INV_b being skewed such that it can barely *interpret* a logical-`1`. Consider the state where the input of INV_a is a logical-`0` and its output (the input of INV_b) is a logical-`1`. A small DC noise can raise the input voltage, thus causing INV_a to no longer *interpret* its input as a logical-`0`, thus resulting in a lowering of its output node voltage. This, in turn, can result in INV_b no longer *interpreting* its input as a logical-`1`, thus resulting in INV_b raising it's output node voltage. This, in turn, pushes INV_a even further away from interpreting its input as a logical-`0`, and so on down the *infinite chain* until the bistable pair 'flips' digital state.

On the other hand, consider an actual linear chain of inverters, as in Figure 3.16. Due to parameter variation, each inverter in the chain behaves differently. Suppose that the chain begins with the identical sequence of skewed INV_a driving a skewed INV_b, but INV_b now drives a different inverter INV_c. Again,

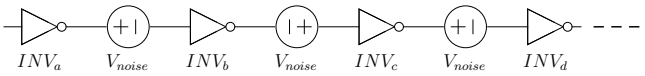


Figure 3.16: Chain of inverters.

consider the same state and event where the input of INV_a is a logical-0 and its output (the input of INV_b) is a logical-1, and a small DC noise raises the input voltage, thus causing INV_a to no longer *interpret* its input as a logical-0, resulting in a lowering of its output node voltage. This, in turn, results in INV_b no longer *interpreting* its input as a logical-1, resulting in INV_b raising it's output node voltage. Suppose, however, that INV_c is a robust inverter and completely restores the rather poor logical-0 generated by INV_B to approximately 0mV. That is, the condition that causes INV_a and INV_b to flip state if configured as a cross-coupled pair may not cause a failure with INV_a and INV_b in a linear chain.

In order to calculate the probability of failure for a chain of inverters, the notion of what it means for a chain to fail must be defined. Not unexpectedly, this definition quickly becomes a problem of logic level *interpretation*, and a clear definition for what it actually means for a chain to fail does not immediately follow, but it is possible to sidestep the problem. That is, cross-coupled inverter static noise margin analysis avoids the definition of failure of an individual inverter by considering a bistable *loop*. Analogously, consider a chain of an even number, $n > 2$, of inverters. If the output of the last inverter in the chain is connected to the input of the first inverter, then the chain becomes a state-holding ring (*loop*). The definition of failure naturally follows as a failure of the ring to maintain state. Informally, the requirement of an even number of stages does not result in a loss of generality, as it is always possible to calculate a tight upper and lower bound on failure rate by considering a chain with one extra and one fewer inverters respectively.[7]

As with the cross coupled inverter analysis, the NM_H and NM_L can be used to generate an upper bound, a lower bound, and an approximation for the probability of failure for chains of inverters. For a chain of inverters, the worst-case, demonic, DC noise consists of alternating positive and negative voltage sources acting contrary to the desired state of each inverter input. That is, if the desired input to a gate is logical-1, *i.e.,* V_{DD}, then a voltage source that acts to lower this electrical potential is said to act contrary to the desired state. In steady-state, a linear chain of n functional inverters consists of alternating sequences of 0 and 1 at the input of each inverter. As such, there are two possible digital states for such a chain: the sequence either begins with a 1 or it begins with a 0. Correspondingly, there are two states for alternating demonic noise sources; the first DC noise source is either positive or it is negative.

Consider a linear chain, CH_a, of n inverters with demonic noise sources, as in Figure 3.16 (with the constraint that n is an even integer greater than 2). The chain is said to fail if the corresponding ring, created by connecting the output of the last inverter to the input of the first inverter, fails to maintain state when all inverter inputs are properly initialized with alternating values of 0 and 1. The chain and ring fail with

[7]The analysis of a ring consisting of an odd number of gates is difficult, because such a ring should oscillate in steady state. The method of static DC analysis used throughout this work is a poor means of modeling an oscillating circuit; further exploration of this problem is left as future work.

respect to a noise margin target, NM_T, when the ring fails to maintain state with demonic noise sources with $V_{noise} = NM_T$ or $V_{noise} = -NM_T$. Given the assumption of statistical independence between the noise margins of different gates pairs (as discussed in Section 3.4.2), the probability of chain failure can be analyzed and computed in terms of the NM_H and NM_L of pairs of gates.

3.4.5.1 Heuristic Upper Bound

Consider a labeling of inverters in the chain CH_a such that the first inverter is labeled as INV_1, the second inverter as INV_2, and so on with the last inverter being INV_n. If the chain fails, then it follows that there exists some inverter pair in the chain, INV_i driving INV_{i+1}, with $NM_H(INV_i, INV_{i+1}) \leq NM_T$ and/or $NM_L(INV_i, INV_{i+1}) \leq NM_T$, which leads to the same probabilistic upper bound for cross-coupled pairs which was discussed in Section 3.4.3.1. For chains, however, this is not a tight upper bound. Empirically, the cross-coupled pair heuristic approximation (see Section 3.4.3.3) leads to a tighter upper bound for chains of gates. Consider two connected pairs of inverters, the set (INV_i, INV_{i+1}, INV_{i+2}); if the chain fails, then it is likely that either (1) $NM_L(INV_i, INV_{i+1}) \leq NM_T$) and $NM_H(INV_{i+1}, INV_{i+2}) \leq NM_T$), and/or (2) $NM_H(INV_i, INV_{i+1}) \leq NM_T$) and $NM_L(INV_{i+1}, INV_{i+2}) \leq NM_T$). With the assumption of statistical independence, the upper bound on the probability of failure for the chain can be approximated as,

$$
\begin{aligned}
& P(FAIL(CH_a, NM_T) \\
\lesssim\ & P\Bigg(\bigcup_{i \in \{1,2,\ldots,n-2\}} \\
& \big((NM_H(INV_i, INV_{i+1}) \leq NM_T + \delta u \cap \\
& \quad NM_L(INV_{i+1}, INV_{i+2}) \leq NM_T + \delta u) \\
& \cup (NM_L(INV_i, INV_{i+1}) \leq NM_T + \delta u \cap \\
& \quad NM_H(INV_{i+1}, INV_{i+2}) \leq NM_T + \delta u) \big) \Bigg)
\end{aligned}
\tag{3.21}
$$

where δu is a small constant used to maintain the boundary over a wide range of NM_T and V_{DD}. The directly computable form of this follows directly from Section 3.4.1.

3.4.5.2 Lower Bound

A heuristic for the lower bound has the same form, but a small constant δl must be subtracted from the NM_T.

$$\begin{aligned}
&P(FAIL(CH_a, NM_T) \\
\gtrsim\ &P\Bigg(\bigcup_{i\in\{1,2,\ldots,n-2\}} \\
&\big((NM_H(INV_i, INV_{i+1}) \leq NM_T - \delta l \cap \\
&\quad NM_L(INV_{i+1}, INV_{i+2}) \leq NM_T - \delta l) \\
&\cup (NM_L(INV_i, INV_{i+1}) \leq NM_T - \delta l \cap \\
&\quad NM_H(INV_{i+1}, INV_{i+2}) \leq NM_T - \delta l)\big)\Bigg).
\end{aligned} \tag{3.22}$$

3.4.5.3 Approximation

As expected, the heuristic approximation follows from the upper and lower bound heuristics.

$$\begin{aligned}
&P(FAIL(CH_a, NM_T) \\
\approx\ &P\Bigg(\bigcup_{i\in\{1,2,\ldots,n-2\}} \\
&\big((NM_H(INV_i, INV_{i+1}) \leq NM_T + \delta \cap \\
&\quad NM_L(INV_{i+1}, INV_{i+2}) \leq NM_T + \delta) \\
&\cup (NM_L(INV_i, INV_{i+1}) \leq NM_T + \delta \cap \\
&\quad NM_H(INV_{i+1}, INV_{i+2}) \leq NM_T + \delta)\big)\Bigg).
\end{aligned} \tag{3.23}$$

Empirically, δu and δl can be defined in terms of δ. For the devices considered in this chapter: INV, NAND2, NOR3, NAND3, NOR3, AOI21, and for noise margin targets between $0\%V_{DD}$ and $20\%V_{DD}$, a relative offset of $3\%V_{DD}$ is sufficient. That is, $\delta u = \delta l = \delta + 3\%V_{DD}$. Figures 3.17, 3.18, and 3.19 depict the upper bound, lower bound, and approximations for a chain of 20 inverters.

Finally, a circuit, C_a, composed of n chains of inverters, is said to fail if any chain fails. That is, with chain labeled as $CH_1, CH_2, ..., CH_n$,

$$\begin{aligned}
&P(FAIL(C_a, NM_T) = \\
&P\left(\bigcup_{i\in\{1,2,\ldots,n\}} FAIL(CH_i, NM_T) \right).
\end{aligned} \tag{3.24}$$

As with the cross-coupled inverter analysis in Section 3.4.4, Equation 3.24 can be re-written in terms of a

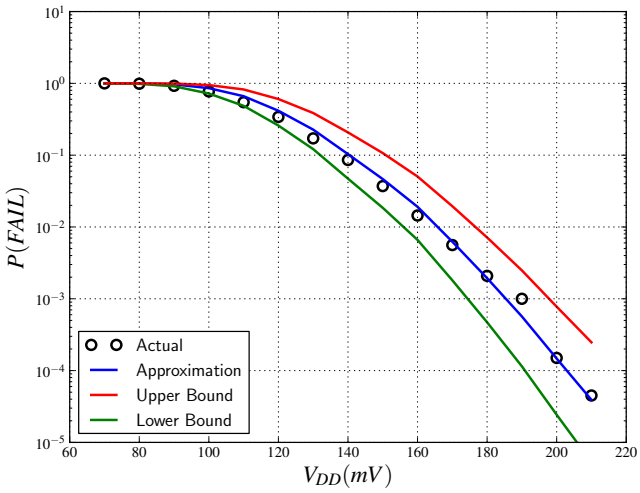


Figure 3.17: Probability of chain of 20 inverters failing with $NM_T = 0\%V_{DD}$ in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner). For the heuristic approximation, the mean absolute error is 17%, and the maximum absolute error is 43% with $\delta = -3.2\%V_{DD}$.

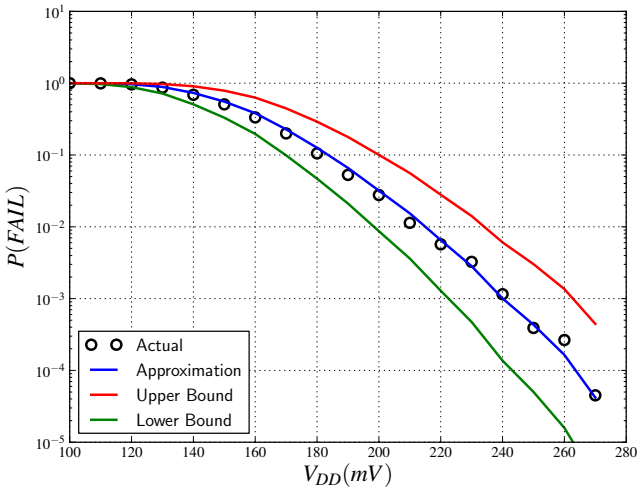


Figure 3.18: Probability of chain of 20 inverters failing with $NM_T = 10\%V_{DD}$ in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner). For the heuristic approximation, the mean absolute error is 13%, and the maximum absolute error is 38% with $\delta = -2.3\%V_{DD}$.

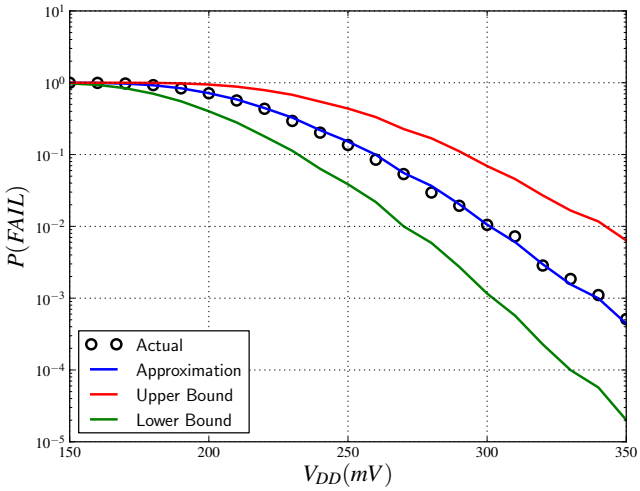


Figure 3.19: Probability of chain of 20 inverters failing with $NM_T = 20\%V_{DD}$ in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner). For the heuristic approximation, the mean absolute error is 6.8%, and the maximum absolute error is 24% with $\delta = -1.8\%V_{DD}$.

global conjunction instead of disjunction as

$$P(FAIL(C_a, NM_T)) = \\ 1 - P\left(\bigcap_{i\in\{1,2,\ldots,n\}} \neg FAIL(CH_i, NM_T)\right). \tag{3.25}$$

Given the assumption of VTC parameter independence between gate pairs, and hence chains (see Section 3.4.2), the global conjunction can be treated as a product, giving a readily computable compact expression for the probability of failure and hence robustness of a circuit consisting of chains of inverters, one of the goals of this section. That is,

$$P(FAIL(C_a, NM_T)) = \\ 1 - P\left(\prod_{i\in\{1,2,\ldots,n\}} \neg FAIL(CH_i, NM_T)\right). \tag{3.26}$$

Using Equation 3.26, Figure 3.20 gives the probability of failure for 2*2e28 independent inverters in the form of chains. As with the cross-coupled pairs, simulation with statistical SPICE is infeasible, so δ is taken from the considerably smaller experiments depicted in Figure 3.19. The probability of failure of chains of inverters is considerably lower than that of cross-coupled pairs (with the same number of devices, noise-margin target, and V_{DD}), as depicted in Figure 3.21. The failure probabilities are similar for cross-coupled pairs of inverters operating $50 - 100$mV above the inverter chain supply voltage. This is the first work to

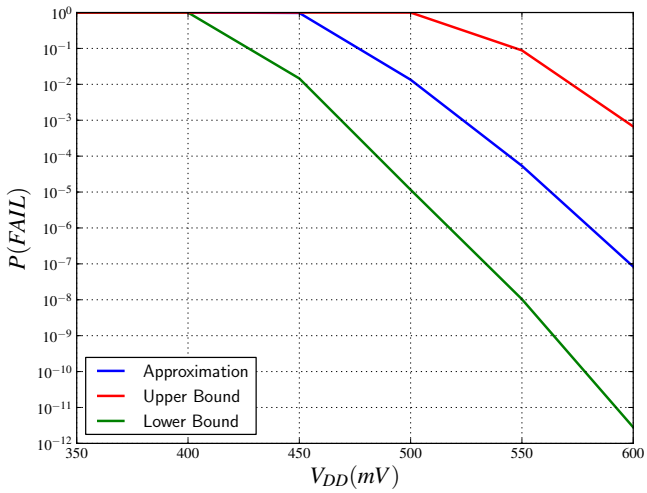


Figure 3.20: Probability of failure for 2*2e28 minimum-size inverters in chains with $NM_T = 20\%V_{DD}$ in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner, and $\delta = -1.8\%V_{DD}$).

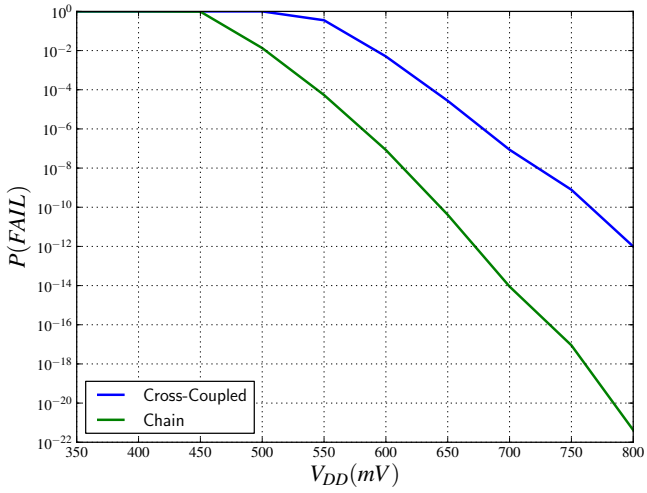


Figure 3.21: Probability of failure for 2*2e28 minimum-size inverters in chains compared to that of 2e28 minimum size cross-coupled pairs with $NM_T = 20\%V_{DD}$ in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner).

quantify and compare these two very different devices' configurations and corresponding probabilities of failure.

3.5 Generalized Circuit Robustness

The goal of this section is to extend the notions of static noise margins to a larger gate set than that of inverters alone. As with the analysis in Section 3.4, the main goal is to generate a composable robustness metric, so that the robustness of an arbitrary network of standard cells can be easily computed.

3.5.1 VTC Parameters of Combinational Gates

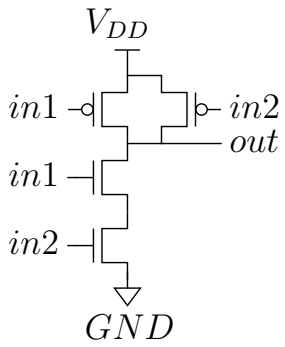


Figure 3.22: NAND2.

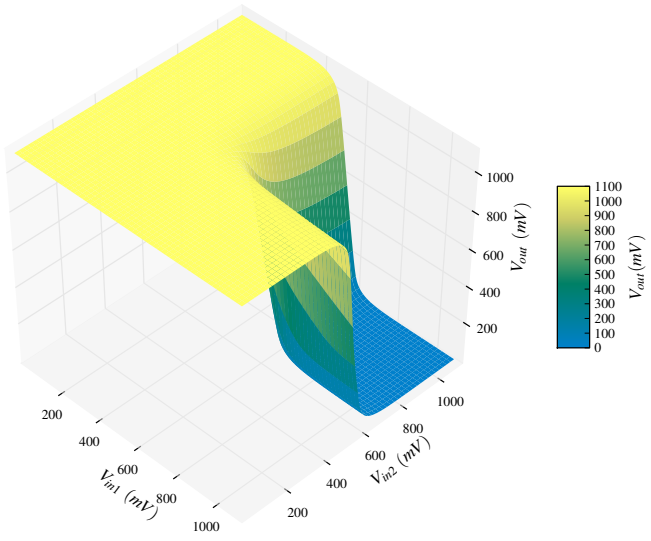


Figure 3.23: Voltage transfer characteristic for the minimum-size NAND2 (depicted in Figure 3.22) in a commercial 40-nm low-power CMOS process ($V_{DD} = 1.1V$, $25°C$, TT-Corner)

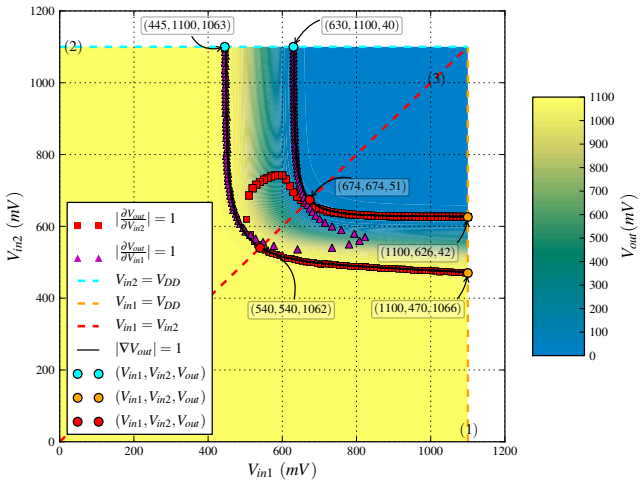


Figure 3.24: Voltage transfer characteristic for the minimum-size NAND2 (depicted in Figure 3.22) in a commercial 40-nm low-power CMOS process ($V_{DD} = 1.1V$, $25°C$, TT-Corner).

Consider a combinational CMOS 2-input NAND gate, NAND2, with nodes labeled as in Figure 3.22. If the input nodes, $in1$ and $in2$, are treated independently, then the VTC describes V_{out} as a function of both V_{in1} and V_{in2}, as depicted in Figures 3.23 and 3.24. Figure 3.23 provides a three dimensional view, and Figure 3.24 plots the VTC in the $V_{in1} \times V_{in2}$ plane with V_{out} encoded by color. The partial derivatives $\frac{\partial V_{out}}{\partial V_{in1}}$ and $\frac{\partial V_{out}}{\partial V_{in2}}$ describe two continuums of unity gain points depicted by purple triangles and red squares, respectively, in Figure 3.24. The gradient is given by

$$\nabla V_{out} = \frac{\partial V_{out}}{\partial V_{in1}}\mathbf{i} + \frac{\partial V_{out}}{\partial V_{in2}}\mathbf{j}, \tag{3.27}$$

where $\mathbf{i}$ and $\mathbf{j}$ are the unit vectors in the $V_{in1} \times V_{in2}$ plane. The two continuums of unity gain points defined by $|\nabla V_{out}| = 1$ (depicted by a black line in Figure 3.24) are analogous to an inverter's two unity gain points given by $|\frac{dV_{out}}{dV_{in}}| = 1$ in Section 3.3.1. In fact, for an inverter the two measures are identical, *i.e.,* $|\nabla V_{out}| \equiv |\frac{dV_{out}}{dV_{in}}|$. Moreover, the magnitude of ∇V_{out} is the most general measure for determining unity gain points, as it is applicable to any gate regardless of the number of inputs.

For noise margin analysis, choosing individual unity gain points as representative approximations is an important simplification, and individual points can be chosen by considering *slices* of the VTC (planes orthogonal to $V_{in1} \times V_{in2}$). Three *slices* of the NAND2 VTC are depicted by dashed lines in Figure 3.24. These three *slices* are of particular interest for two reasons. First, they give the upper and lower unity-gain bounds in terms of V_{in1} and V_{in2}. Second, they correspond to a logical reduction of the NAND2 to that of an inverter. That is, if either input is tied to logical-1 or if both inputs are tied together, then the NAND2 is functionally equivalent to an inverter. In Figure 3.24 the three possible inverter-equivalent slices are depicted

by: (1) an orange dashed-line corresponding to tying $in1$ to V_{DD} and sweeping $in2$ from GND to V_{DD}, (2) a light-blue dashed-line corresponding to tying $in2$ to V_{DD} and sweeping $in1$ from GND to V_{DD}, and (3) a red dashed-line generated by tying $in1$ to $in2$ and sweeping them together.

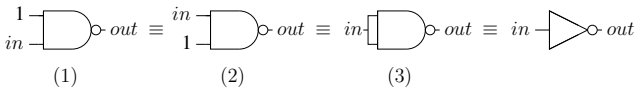


Figure 3.25: NAND2 inverter equivalence.

The idea of inverter-equivalence is general and can be used to generate the boundary unity-gain points for arbitrary gates. Consider an inverting binary CMOS gate, G, with n inputs and a single output. Gate G can be made to act logically as a single input/output inverter for some assignment of inputs where inputs can be tied together, tied to `1`, or tied to `0`. Since G is an inverting CMOS gate, one or more inverter-equivalent input assignments exist, and the assignments depend on the topology of G. The three inverter-equivalent slices from Figure 3.24 are depicted at the gate level in Figure 3.25.

A general notion of an inverter equivalent assignment is helpful. Let G be an inverting binary CMOS gate with k inputs labeled as $in1, in2, ..., ink$, a single output, out, and with functionality defined by $V_{out} = G(V_{in1}, V_{in2}, ..., V_{ink})$. The set of inverter equivalent input assignments to G, denoted $IE(G)$, is a set of k-tuples, $(ie_1, ie_2, ..., ie_k)$, where $ie_i \in (\texttt{1}, \texttt{0}, \texttt{in})$, and $(ie_1, ie_2, ..., ie_k) \in IE(G)$ if and only if $G(ie_1, ie_2, ..., ie_k)$ is functionally equivalent to an inverter with input in, and output out. For the NAND2, the three inverter equivalent input assignments are (1) $(\texttt{1}, \texttt{in})$, (2) $(\texttt{in}, \texttt{1})$, and (3) $(\texttt{in}, \texttt{in})$. In order to work with inverter equivalent input assignments, it is convenient to define F, a simple mapping function between real voltages and elements of $(\texttt{1}, \texttt{0}, \texttt{in})$. That is,

$$F(V_i) = \begin{cases} \texttt{0} & \text{if } V_i = GND \\ \texttt{1} & \text{if } V_i = V_{DD} \\ \texttt{in} & \text{otherwise.} \end{cases} \tag{3.28}$$

With a notion of inverter equivalence, it is possible to define a representative set of unity gain points for a gate. Let G be an inverting binary CMOS gate with k inputs labeled as $in1, in2, ..., ink$, and a single output, out. The gradient of V_{out} is defined as

$$\nabla V_{out} = \frac{\partial V_{out}}{\partial V_{in1}}\mathbf{i}_1 + \frac{\partial V_{out}}{\partial V_{in2}}\mathbf{i}_2 + ... + \frac{\partial V_{out}}{\partial V_{ink}}\mathbf{i}_k, \tag{3.29}$$

where $\mathbf{i}_1$, $\mathbf{i}_2$, ..., $\mathbf{i}_k$ are the corresponding unit vectors. The *representative set of unity gain points* for G,

$RS(G)$, is defined such that

$$(V_{in1}, V_{in2}, ..., V_{ink}, V_{out}) \in RS(G) \text{ if and only if}$$
$$|\nabla V_{out}(V_{in1}, V_{in2}, ..., V_{ink})| = 1 \text{ , and}$$
$$(F(V_{in1}), F(V_{in2}), ..., F(V_{ink})) \in IE(G) \text{ , and}$$
$$\text{for all } x, y \in (1, 2, ..., k) \tag{3.30}$$
$$\text{if } F(V_{inx}) = \texttt{in} \text{ and } F(V_{iny}) = \texttt{in} \text{ then } V_{inx} = V_{iny}. \tag{3.31}$$

For the NAND2, the representative set of unity gain points are depicted in Figure 3.24 as light-blue, orange, and red circles with annotated values. The values for each slice $(V_{in1}, V_{in2}, V_{out})$ are (1): (1100,470,1066), (1100,626,42) (2): (445,1100,1063), (630,1100,40), (3): (540,540,1062), (674,674,51). These representative points can be mapped back to simple pairs of the form (V_{in}, V_{out}) by using the inverter equivalent input assignment to remove the references to V_{DD}, GND, and shared inputs. For the NAND2 this *reduced* representative set of unity gain points is (1): (470,1066), (626,42) (2): (445,1063), (630,40), (3): (540,1062), (674,51).

Finally, the VTC parameters can be defined using the reduced set of unity gain points. The usual mapping of unity gain points to VTC parameters can be employed, so for the NAND2, (1): $V_{IL} = 470$, $V_{OH} = 1066$, $V_{IH} = 626$, $V_{OL} = 42$, (2): $V_{IL} = 445$, $V_{OH} = 1063$, $V_{IH} = 630$, $V_{OL} = 40$, and (3): $V_{IL} = 540$, $V_{OH} = 1062$, $V_{IH} = 674$, $V_{OL} = 51$. Statistical analysis is greatly simplified when a single set of representative VTC parameters is chosen, but the parameters—as measured with (1), (2), and (3)—differ. It is clear that V_{OH} and V_{OL} are nearly constant; this is expected (see Section 3.4.1). The two inverter equivalent input assignments where a single input is tied to V_{DD} ((1) and (2)) are highly symmetric and have only slightly different values for V_{IH} and V_{IL}, respectively. The input assignment wherein both inputs are tied together, (3), does differ significantly in terms of V_{IH} and V_{IL} from (1) and (2).

Consider the measurement of V_{IL} performed by sweeping the input(s) from GND to V_{DD} using (1) as compared to (3). The value of V_{IL} corresponds to the greatest input voltage that still results in the output being pulled-up to a logical-1. The NAND2 contains 2 parallel PFETs with gates connected to $in1$ and $in2$, respectively. With input assignment (1), $in1$ is tied to V_{DD}, causing the corresponding PFET to effectively turn off, *i.e.,* it contributes only sub-threshold leakage current to the pull-up network as $in2$ is swept from GND to V_{DD}. As V_{in2} is increased, the corresponding PFET begins to turn off and the NFETs begin to turn on, thus transitioning the output towards a logic-0; V_{IL} is the input voltage at which this transition occurs. With (3), both inputs are tied together, and the parallel PFETs actively pull up the output node together as the input is swept. The parallel PFETs in (3) continue to actively pull up the output node as the input voltage is increased beyond the V_{IL} from (1). As such, V_{IL} as measured with (3) is greater than V_{IL} as measured with (1). An analogous, but reciprocal explanation can be given for V_{IH}. That is, V_{IH} as measured with (1) is greater than V_{IH} as measured with (3).

In order to provide an upper bound on the robustness, the representative set of VTC parameters should be chosen so as to overestimate the probability of failure of a gate. This corresponds to underestimating both NM_H and NM_L; this, in turn, necessitates underestimating V_{IH} and overestimating V_{IL}.[8] Since V_{OH} and V_{OL} are approximately constant across inverter equivalent input assignment slices, the smallest V_{IH} and the largest V_{IL} should be chosen from the *reduced* representative set of unity gain points. For the NAND2 this corresponds to selecting the VTC parameters from different slices: V_{IH} from (1) and V_{IL} from (3).[9] This somewhat complicates the task of gate characterization, so in this chapter the VTC parameters from (1) are chosen as the representative set, despite the fact that this simplifying choice slightly underestimates V_{IL}. In terms of calculating the probability of failure, this simplification has little impact.

Finally, the VTC parameters for an arbitrary gate can be defined. Let G be an inverting binary CMOS gate with m inputs and a single output, out, and let $RS(G)$ be the *reduced representative set of unity gain points* for G. Assume that $RS(G)$ has cardinality n, and elements labeled as (V_{in_i}, V_{out_i}) for $i \in \{1, 2, ..., n\}$. The VTC parameters for G are defined as

$$\begin{aligned} V_{IH}(G) &= min(V_{in_i}) \\ V_{IL}(G) &= max(V_{in_j}) \\ V_{OH}(G) &= min(V_{out_k}) \\ V_{OL}(G) &= max(V_{out_l}), \end{aligned}$$

for $i, j, k, l \in \{1, 2, ..., n\}$.

3.5.2 Statistical Noise Margins of Combinational Gates

In order to give general definitions for NM_H and NM_L, it is necessary to consider the input VTC parameter correlation between multiple inputs of the same gate. As shown in Figure 3.26, the input VTC parameters are highly uncorrelated over a wide range of V_{DD}. Given this and the treatment of output VTC parameters as regular variables, arbitrary networks of combinational gates with fan-in and fan-out greater than unity can be broken apart into equivalent gate-pairs, and ultimately, into inverter-equivalent pairs for statistical analysis; *i.e.,* for the purpose of computing the probability of circuit failure.

Consider a circuit, C_a, composed of a network of n combinational gates in an array of simple linear chains; C_a is said to fail if any chain of gates within the circuit fails (see Equation 3.24). In general, digital circuits consist of networks of combinational gates organized as interconnecting chains, wherein some gates drive multiple gates, some gates are driven by multiple gates, or both. Consider a circuit, C_b, composed of a network of n combinational gates with interconnecting chains, *i.e.,* some gates within C_b drive multiple gates

[8] Assuming that the corresponding variances are approximately equal.

[9] In a similar fashion, the smallest V_{OH} and largest V_{OL} could be chosen as representative VTC parameters; however, the output VTC parameters are approximately constant, so they can also be chosen arbitrarily or by convenience.

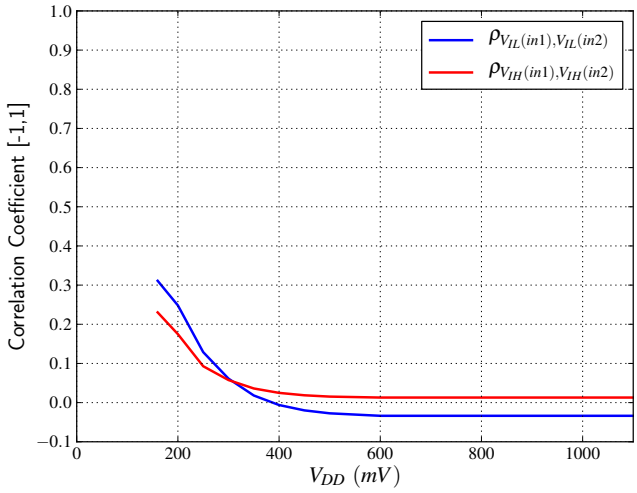


Figure 3.26: NAND2 input Correlation in a commercial 40-nm low-power CMOS process ($V_{DD} = 1.1V$, $25°C$, TT-Corner).

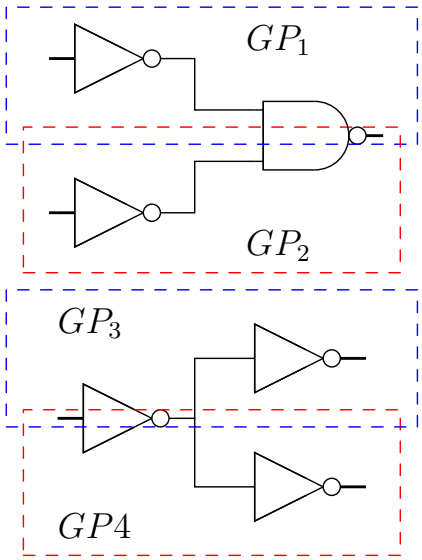


Figure 3.27: Equivalent gate-pairs formed from multiple fan-in and fan-out gate networks. GP_1 and GP_2 are formed for each input of the NAND gate, and GP_3 and GP_4 are due to the inverter fan-out.

and some gates have multiple inputs. Consider a gate, $G_x \in C_b$ that drives k gates in C_b, labeled as G_1, G_2, ..., G_k. Since the output VTC parameters of G_x are not stochastic in nature, these gates can be treated as k equivalent gate-pairs (G_x, G_i) for $i \in \{1, 2, ..., k\}$. As an example, consider GP_3 and GP_3 in Figure 3.27. Similarly, consider k gates in C_b, labeled as G_3, G_4, ..., G_k that drive (fan-in) to a multi-input gate, $G_x \in C_b$. Given that the input VTC parameters of G_x are independent, each pair, (G_i, G_x) for $i \in \{1, 2, ..., k\}$, can be considered as components of independent equivalent gate-pairs, as illustrated in 3.27 by GP_1 and GP_2. Finally, as discussed in Section 3.5.1, every equivalent gate-pair can be analyzed as an inverter equivalent pair, and the robustness of a circuit consisting of arbitrary connections of combinational gates can be computed by way of the methods detailed in Section 3.4.5. Figure 3.28 plots the failure probabilities for a linear chain of 20 gates: alternating NAND2 and NOR2 with $NM_T = 10\%V_{DD}$. Similarly, Figure 3.29 shows the failure probabilities for a chain consisting of alternating NAND3 and NOR3 gates with $NM_T = 20\%V_{DD}$.

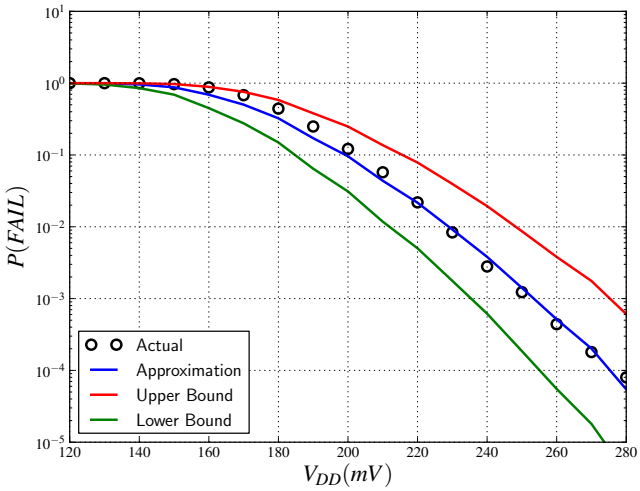


Figure 3.28: Probability of chain of 20 combinational gates failing (the chain consists of alternating NAND2, NOR2 gates) with $NM_T = 10\%V_{DD}$ in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner). For the heuristic approximation, the mean absolute error is 16%, and the maximum absolute error is 36% with $\delta = -1.2\%V_{DD}$.

3.5.3 Applications

The methods presented in this chapter give a circuit designer the ability to calculate the robustness of a digital circuit composed out of gates. That is, for some circuit, C_a, and a target noise margin, NM_T, Equation 3.26 gives the probability that some gate chain in C_a has a noise margin less than the target, *i.e.,* a probability of failure $P(FAIL)$. This quantity can, of course, instead be considered as a passing probability $P(PASS)$ by subtracting it from unity, and this passing probability can be thought of as a parametric yield. That is, if $P(PASS) = 95\%$, then in 95% of instances of C_a, all gates will exceed the noise margin target constraint

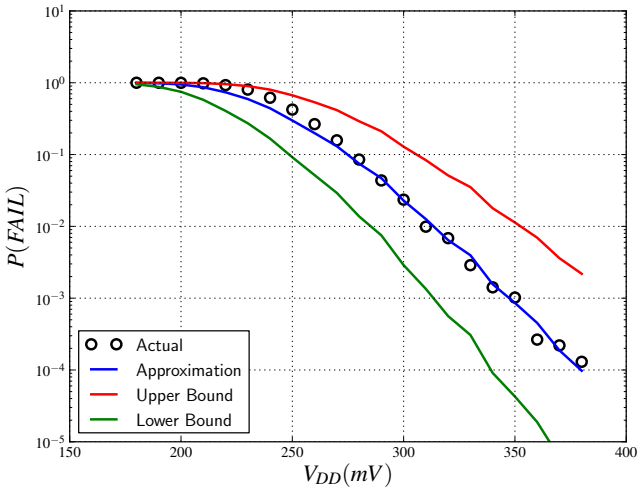


Figure 3.29: Probability of chain of 20 combinational gates failing (the chain consists of alternating NAND3, NOR3 gates) with $NM_T = 20\%V_{DD}$ in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner). For the heuristic approximation, the mean absolute error is 16%, and the maximum absolute error is 67% with $\delta = -1.3\%V_{DD}$.

(this is definitionally a parametric yield). If the circuit under consideration, C_a, is an entire microprocessor, then this parametric yield can be included as a part of the die yield calculation.

In Equation 3.26, the circuit under consideration, C_a, is an independent variable, and $P(FAIL)$ is a dependent variable. It is straightforward to instead treat C_a as dependent on $P(FAIL)$. In this way, a designer can choose a NM_T and a yield, and then calculate the maximum number of gates that satisfy this constraint (*i.e.,* how large of a circuit can be built). Figure 3.30 plots the maximum number of equivalent gate-pairs that satisfy a NM_T and yield constraint vs. V_{DD}. It is clear from this figure that the gate choice has only a small impact on how large of circuit can be constructed, and the most important constraint is supply voltage; *i.e.,* the maximum circuit size is exponential in V_{DD}. Figure 3.31 plots the maximum NM_T that can be guaranteed (for 1M equivalent gate-pairs in chains and a yield of 95%) versus V_{DD}.

3.6 Related Work

The earliest works to consider digital circuit robustness with respect to noise and a definition of a static noise margin come independently from Lo and Hill, respectively [46, 47, 55]. More recently, Shepard proposed a systematic approach to incorporating noise margins into the design process of large circuits via Harmony (an EDA tool) [84]. The primary problem with Harmony is that it does not account for parameter variation, so it is not sufficient for modern low-voltage circuit analysis. It may be possible to apply the robustness metrics and computation techniques detailed in this chapter to a tool like Harmony, but this is left as future work.

Noise margin based analysis of memory cells [81] is common, and a number of works consider the

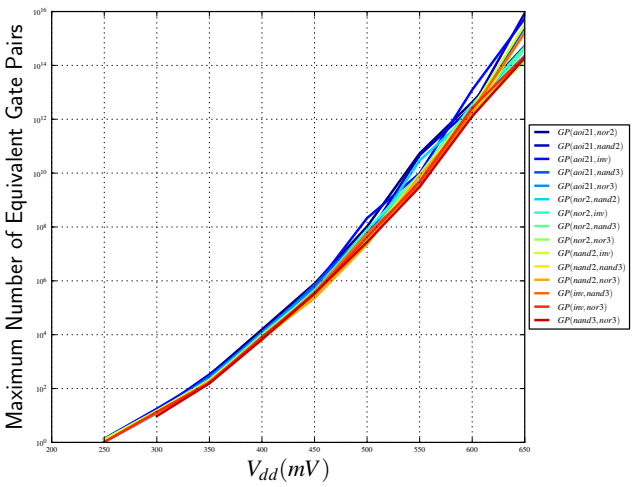


Figure 3.30: Maximum number of equivalent gate-pairs vs. V_{DD} with $NM_T = 20\%V_{DD}$ and $yield = 95\%$ in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner). Chains consist of alternating gates, and all combinations from the set $(INV, NAND2, NOR2, AOI21, NAND3, NOR3)$ are considered.

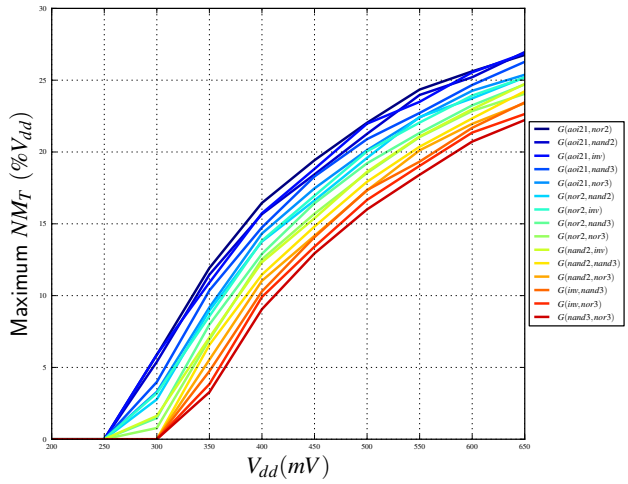


Figure 3.31: Maximum NM_T vs. V_{DD} for $1M$ equivalent gate-pairs and $yield = 95\%$ in a commercial 40-nm low-power CMOS process ($25°C$, TT-Corner). Chains consist of alternating gates, and all combinations from the set $(INV, NAND2, NOR2, AOI21, NAND3, NOR3)$ are considered.

effects of parameter variation in SRAM based analysis, *e.g.,* [13, 19, 20, 41, 65, 82, 95]. These works perform statistical analysis using the SNM expression (Equation 3.3) and are thus limited by the *min* function. Calhoun works around this somewhat in [20] by using order statistics on the tail of the SNM distribution. A few works (*e.g.,* [14, 15, 53, 54]) analyze combinational gate failures due to parameter variation when operating at low voltages, but these works only consider the single case where all inputs of each gate are tied together. Moreover, these works only consider a simple binary failure model (*i.e.,* $SNM > 0$ or $SNM < 0$), as opposed to the generalized noise margin target based analysis presented in this chapter.

Several works specifically consider and model some of the effects of parameter variation on circuits operating sub-threshold. Chen considers the limitations in terms of large fan-ins and fan-outs in [24], and Pu [74] uses affine arithmetic to model the effects of parameter variation on V_{OH} and V_{OL}. Alioto derives an accurate closed-form sub-threshold SNM model in [3] and considers the effects of variation on sub-threshold circuits by way of analyzing the imbalance factor (IF) between the PFET and NFET networks that make up a gate. Some of the work discussed in this chapter was initially presented in [52].

3.7 Conclusion

This chapter presents a metric for digital circuit robustness with respect to parameter variation and noise. The robustness metric is general, and while only applied to CMOS circuits in this chapter, can be extended to other technologies (possible future work). Additionally, a compact method for calculating the robustness of CMOS circuits operating sub-threshold or near-threshold is detailed and validated. The method of calculation relies on a new compact representation of parameter variation at the cell level; as such, the robustness of an extremely large circuit can be quickly, efficiently, and accurately computed. The statistical details of the model are flushed out and validated against SPICE simulations of foundry provided statistical BSIM simulations in a modern (40-nm) technology. This work relies extensively on the notion of a static noise margin (see Section 3.3.1). This notion, previously defined exclusively for cross-coupled gate-pairs, is extended in three important ways:

- it is turned into a statistical quantity in Section 3.4.2,
- it is extended to cover chains of gates in Section 3.4.5, and
- it is generalized for use with any inverting single-output CMOS gate in Section 3.5.1.

As with all metrics, there are limitations to the applicability of the work presented in this chapter. Many of the calculations rely on the assumption of statistical independence of parameter variation between different gates. This assumption is discussed in detail and justified in Section 3.4.2. If this assumption does not hold, the effects of correlation can be accounted for by way of adding a correlation coefficient to Equations 3.16, 3.20, and 3.26. These effects are likely well modeled as spatial correlation [89], so accurate correlation models may require knowledge of circuit layout. Quantifying these effects, which are currently only significant

at high supply voltages, is left as future work. Finally, choosing different noise margin targets for different gates is left as future work (circuit noise can vary from gate to gate).

Chapter 4

A Necessary and Sufficient Timing Assumption for Speed-Independent Circuits

4.1 Introduction

Asynchronous logic can be effectively engineered within a variety of different frameworks, *e.g.,* via quasi-delay insensitive (QDI) [58, 59, 62] or speed-independent (SI) circuits [64, 66]. Across the various frameworks there are clearly many important differences, but these frameworks also share certain issues that seem to be inherent to asynchronous circuit design; in particular, the notion of *forks*. For example, the class of SI circuits is characterized as the set of circuits that are functionally correct regardless of gate and wire delays, *except* at forks; similarly, forks play a crucial role in the context of QDI circuits. In fact, if no delay assumptions are made about forks, then the resulting delay-insensitive (DI) circuits are extremely limited in functionality [60]. Moreover, for many asynchronous logic frameworks, the relative delays through fork branches form the basis of all timing assumptions and the corresponding timing closure. This suggests that in order to gain insight into the exact similarities and differences between these frameworks, it may be fruitful to compare their timing assumptions.

Making such comparisons can be difficult because different frameworks use different terms and mathematical constructions; a mathematical setting in which common terms are used to describe all of the timing assumptions is required. Therefore, towards clarifying the nature of forks across multiple asynchronous circuit frameworks, this chapter first formally defines a notion of asynchronous computation and then upon that defines a set of well-known fork-related timing assumptions. Using this foundation, the chapter then proves that one such assumption, the *adversary path timing assumption* [62], is necessary and sufficient for proper SI circuit operation.

The foundation starts with the structure given by *production rule sets* (PRS). This is not crucial: *any number of systems can be used instead*. However, PRS are structured in a way that clearly exposes how forks

and hazards [5, 64] interact. Moreover, PRS can be used to model arbitrary switching networks, and therefore can be used to examine important properties of circuits generated within a wide range of asynchronous design methodologies. In addition, the proof is just one example of how the formalization can be used. It can also serve as a foundation for other proofs or definitions, and it can even be used as a basis for computerized proofs.

Each of the following technical sections contains both an informal overview of main concepts, with examples derived from the circuit depicted in Figure 4.1 (a closed variant of a circuit used in [60]), as well as thoroughly developed mathematical details. The details add rigor and some subtle insights, but the main ideas and notation are presented at a higher level. The organization of the chapter is as follows. Section 4.2 reviews production rule sets and defines a set of structural constraints that are assumed throughout. Section 4.3 formally defines a notion of computation with respect to PRS. Then, upon this notion of computation, Section 4.4 defines the relevant timing assumptions for DI, QDI, and SI circuits. The proof and a discussion of its implications are given in Section 4.5. Section 4.6 reviews related work, and Section 4.7 concludes with a summary and a discussion of several assumptions and limitations of this work.

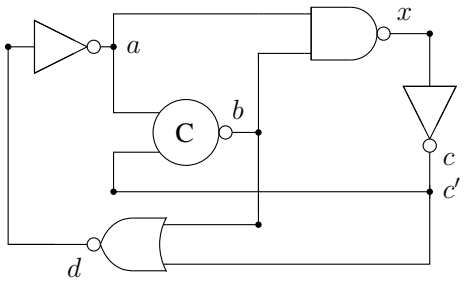


Figure 4.1: Closed simple buffer.

4.2 PRS Structural Constraints

Section 4.2.1 begins by reviewing the traditional definition of PRS [59]. Section 4.2.2 adds a set of structural constraints that facilitate the definition of a formal notion of computation and timing assumptions in Sections 4.3 and 4.4, respectively. These structural constraints ultimately result in a set of "legal" production rule sets, which are called *proper*.

4.2.1 PRS

This section reviews a few of the basic terms from [59]. The mapping from PRS to CMOS transistor networks, used in a number of figures throughout this chapter, also comes from [59]. Additionally, in order to simplify the exposition, this chapter assumes a fixed set V of variables from which the PRS draw node names; $T_{\mathbb{B}}(V)$ denotes the set of Boolean expressions over variables V.

Definition 4.2.1. A *production rule* is any triple

$$(g, x, d) \in T_{\mathbb{B}}(V) \times V \times \{\uparrow, \downarrow\}$$

and is typically denoted $g \mapsto xd$.

Definition 4.2.2. A *production rule set* (PRS) is any finite set of production rules.

The basic intuition for a production rule $g \mapsto x\uparrow$ is that g is a *sufficient* condition to enable the pull-up network in the gate associated with x. This means that the entire condition for the pull-up network can be spread out across multiple production rules $g_i \mapsto x\uparrow$. For example, one of the structural constraints that this chapter enforces, without loss of generality, is that each variable x is defined by *exactly two production rules*, $g^+ \mapsto x\uparrow$ and $g^- \mapsto x\downarrow$.

4.2.2 "Proper" PRS

The definitions in this section serve to impose extra structure on PRS. This added structure facilitates a straightforward definition of computation in Section 4.3 and the mapping of computation to physical circuits; furthermore these constraints simplify the definition of timing assumptions in Section 4.4 and the proof in Section 4.5.

Definition 4.2.3. Let $\mathcal{P}$ be a PRS and $x \in V$, the x *operator on* $\mathcal{P}$, denoted $\mathcal{O}_x$, is defined such that for all $g \mapsto x'd \in \mathcal{P}$

$$g \mapsto x'd \in \mathcal{O}_x \Leftrightarrow x' = x.$$

Definition 4.2.4. Let $\mathcal{P}$ be a PRS. $\mathcal{P}$ *has simple operators* if and only if all $\mathcal{O}_x$ are such that

$$\mathcal{O}_x = \{g^+ \mapsto x\uparrow, g^- \mapsto x\downarrow\},$$

and g^+ and g^- are in *disjunctive normal form*.

Definition 4.2.5. Let $\mathcal{P}$ be a PRS with simple operators. $\mathcal{O}_y$ is called a *wire* if and only if

$$\mathcal{O}_y = \{x \mapsto y\uparrow, \neg x \mapsto y\downarrow\}$$

for some $y \in V$. An operator $\mathcal{O}_x$ is called a *gate* if it is not a wire.

The majority of structural constraints for a proper PRS simply enforce a regular forking structure. These requirements essentially guarantee a one-to-one correspondence between the branches of a fork and *wire operators*. In order to define and force these and other structural requirements, it is necessary to have a clean way of expressing variable sharing within and between the operators, as such sharing can imply forking. This

is formalized beginning with the function π, which counts the number of occurrences of a specified variable in the guard of a production rule.

Definition 4.2.6. $\pi : V \times T_{\mathbb{B}}(V) \longrightarrow \mathbb{N}$:

$$\begin{aligned}
\pi(x, x) &= 1 \\
\pi(x, x') &= 0 \qquad \textit{if } x' \neq x \\
\pi(x, g_1 \wedge g_2) &= \pi(x, g_1) + \pi(x, g_2) \\
\pi(x, g_1 \vee g_2) &= \pi(x, g_1) + \pi(x, g_2) \\
\pi(x, \neg g_1) &= \pi(x, g_1).
\end{aligned}$$

Using Definition 4.2.6, PRS variables are related to each other by constructing a directed multi-graph with a node for every variable and a directed weighted-edge between pairs of variables.

Definition 4.2.7. Let $\mathcal{P}$ be a PRS. Associate to $\mathcal{P}$ a directed multi-graph $(V, E : V \times V \longrightarrow \mathbb{N})$ where

$$E(x, x') = \sum_{g \mapsto x' d \in \mathcal{O}_{x'}} \pi(x, g).$$

The most important information encoded by this graph is a matching of input variables to the output variable of each gate. This is expressed via the following $\longrightarrow$ relation.

Definition 4.2.8. Let $\mathcal{P}$ be a PRS. With respect to $\mathcal{P}$, $\longrightarrow \subseteq V \times V$ is a binary relation defined such that for all x, x'

$$x \longrightarrow x' \Leftrightarrow E(x, x') > 0.$$

Figure 4.2 expands the NAND gate from Figure 4.1 and adds two wires, $\mathcal{O}_{a'}$ and $\mathcal{O}_{a''}$. This expansion illustrates several definitions, *e.g.,* $E(a'', x) = E(a, x) = 1$, $E(b, x) = 2$, $a \longrightarrow a'$, and $a' \longrightarrow a''$. The $\longrightarrow$ relation is employed extensively, and in many cases the following notational conventions are used: $\cdot \longrightarrow x'$, which means the *set* $\{x \mid x \longrightarrow x'\}$, and $x' \longrightarrow \cdot$, which means the *set* $\{x \mid x' \longrightarrow x\}$. With respect to Figure 4.2, $\cdot \longrightarrow x = \{a, a'', b\}$, and $b \longrightarrow \cdot = \{x\}$. This notion usefully extends to multiple arrows and multiple dots; *e.g.,* $x \longrightarrow \cdot \longrightarrow x'$ or $\cdot \longrightarrow \cdot \equiv \longrightarrow$.

The existence of composed wires is equivalent to the statement that there exist wires $\mathcal{O}_y$ and $\mathcal{O}_{y'}$, $y \neq y'$, such that $y \longrightarrow y'$, *e.g.,* in Figure 4.2, $a' \longrightarrow a''$. This sort of composition is not allowed in a proper PRS, and, similarly, gate-to-gate connections are also disallowed, *e.g.,* in Figure 4.1, the variable a acts as both the output of the inverter and an input of the NAND gate. Therefore, between gates, a signal must go through *exactly one* wire.

Definition 4.2.9. Let $\mathcal{P}$ be a PRS. $\mathcal{P}$ has *no wire-to-wire connections* if and only if for all pairs of wires $\mathcal{O}_y, \mathcal{O}_{y'}$, $(y, y') \notin \longrightarrow$.

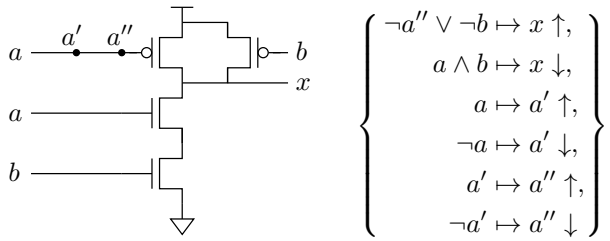


Figure 4.2: CMOS NAND gate and wires.

Definition 4.2.10. Let $\mathcal{P}$ be a PRS. $\mathcal{P}$ has *no gate-to-gate connections* if and only if for all pairs of gates $\mathcal{O}_x, \mathcal{O}_{x'}$, $(x, x') \notin \longrightarrow$.

This leaves the possibility of implicit forking through sharing of wire variables across different gates, which can be removed by enforcing that $\mathcal{P}$ has *explicit inter-operator forks*.

Definition 4.2.11. Let $\mathcal{P}$ be a PRS satisfying the conditions of Definitions 4.2.9 – 4.2.10. $\mathcal{P}$ has *explicit inter-operator forks* if and only if for all wires $\mathcal{O}_y$, $|y \longrightarrow \cdot| = 1$.

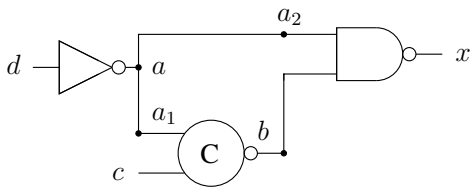


Figure 4.3: Explicit inter-operator fork.

Figure 4.3 transforms an implicit inter-operator fork from Figure 4.1 into an explicit inter-operator fork by connecting inverter $\mathcal{O}_a$ to two new wires $\mathcal{O}_{a_1}$ and $\mathcal{O}_{a_2}$. Nevertheless, variable sharing can occur within a gate, *e.g.,* this happens in Figure 4.2, where $E(b, x) = 2$. This leads to the final structural constraint on forks.

Definition 4.2.12. Let $\mathcal{P}$ be a PRS satisfying the conditions of Definitions 4.2.9 – 4.2.10. $\mathcal{P}$ has *explicit intra-operator forks* if and only if for all wires $\mathcal{O}_y$, if $y \longrightarrow x$ then $E(y, x) = 1$.

Figure 4.4 further expands Figure 4.3 by making explicit the NAND gate intra-operator forks. This necessitates the addition of another new wire, $\mathcal{O}_{a_3}$, to inverter $\mathcal{O}_a$ and two new wires $\mathcal{O}_{b_1}$ and $\mathcal{O}_{b_2}$ to C-element $\mathcal{O}_b$. Definitions 4.2.9 – 4.2.12 ensure that at the switch level, there is a one-to-one correspondence between gate interconnections and wires.

Making *inter-operator* forks explicit is commonplace and essential for a discussion of asynchronous circuits. Making *intra-operator* forks explicit is less typical but not unprecedented (see [73]); they are exposed for completeness in Section 4.4 on timing assumptions. In what follows, a properly structured PRS is closed and is considered to have all of the above properties.

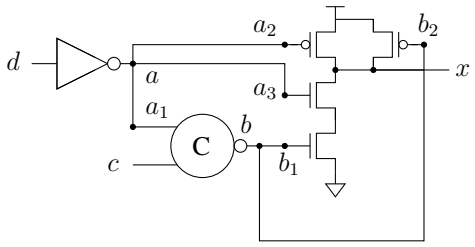


Figure 4.4: Explicit intra-operator fork.

Definition 4.2.13. Let $\mathcal{P}$ be a PRS. $\mathcal{P}$ *is closed* if and only if for all $x \in V$; $\cdot \longrightarrow x \neq \emptyset$ and $x \longrightarrow \cdot \neq \emptyset$.

Definition 4.2.14. Let $\mathcal{P}$ be a PRS. $\mathcal{P}$ *is proper* if and only if $\mathcal{P}$ satisfies the conditions of Definition 4.2.4 and Definitions 4.2.9–4.2.13.

4.3 PRS Semantics

This section defines a mapping from PRS to legal *computations*, where *computations* are legal if they fall within the set of dynamic behaviors defined by a circuit. The formalization treats PRS as a set of concurrent processes with each gate and wire acting individually. The main ideas as well as several examples are presented in Section 4.3.1, followed by further details in Section 4.3.2.

4.3.1 Overview

Conceptually, the definition of computation given in this section treats gates and wires as *independent processes*. These processes are continuously sensitive to the current state of all nodes named in the guards of the corresponding pair of production rules. For example, the expanded NAND gate shown in Figure 4.4 is treated as a process that is sensitive to the state of four nodes: a_2, a_3, b_1, b_2. At any given "step" in the computation, each process can either (a) act on the current state of its inputs by transitioning its target node appropriately, or (b) delay a pending transition to a future step. There is also a third possibility, (c): a gate can express a pending *hazard*.

Ignoring hazards for the moment, the *state* of the circuit nodes is encoded as a function, $\chi : V \longrightarrow \{\mathtt{F}, \mathtt{T}\}$, which maps nodes to logical values. For example, $\chi(a_2) = \mathtt{T}$ means that the current state of node a_2 acts as logical true. Now, consider again the NAND gate and a state

$$\chi(a_2) = \mathtt{T}, \chi(a_3) = \mathtt{T}, \chi(b_1) = \mathtt{T}, \chi(b_2) = \mathtt{T}, \chi(x) = \mathtt{T}.$$

A computation "step" takes the *current state*, χ, to a *new state*, χ'. Corresponding to cases (a) and (b) above, there are two alternatives for x in χ'. Either (a) the pending transition gets expressed and $\chi'(x) = \mathtt{F}$, or (b) the transition is delayed and $\chi'(x) = \chi(x) = \mathtt{T}$.

For every gate, there is also a specific set of undesirable states that expose its non-digital and non-atomic nature. These hazardous states generate uncertainty in the logic value of the gate output. As such, subsequent gates may individually interpret the value as either T, F, or undefined. These possibilities necessitate further enrichment of state beyond $\chi : V \longrightarrow \{\mathtt{F}, \mathtt{T}\}$. First, for hazards to be explicitly manifested, the co-domain of χ is expanded to the set $\{\mathtt{F}, \mathtt{X}, \mathtt{T}\}$, so that χ becomes a function $\chi : X \longrightarrow \{\mathtt{F}, \mathtt{X}, \mathtt{T}\}$. Second, the state is enriched so that it becomes a pair (χ, I) with χ as above and $I \subseteq V$, where I is a set containing all nodes with pending hazards. That is, $x \in I$ implies that case (c) is a valid option, so $\chi'(x) = \mathtt{X}$ is possible in some future computation step.

4.3.2 Formalization

The definition of computation is given as a binary relation on *execution states*. From this point onward, denote by $\mathbb{B}$ the structure with elements $\{\mathtt{F}, \mathtt{X}, \mathtt{T}\}$ and functions $\neg, \wedge, \vee$.

Definition 4.3.1. An *execution state* is any pair

$$(\chi : V \longrightarrow \mathbb{B}, I \subseteq V).$$

In order to define inference rules for generating the next system state χ' from the current state χ, it is useful to have formal notions describing the current "state" of a gate. Intuitively, if χ is such that a gate is being *pulled up*, then the gate is allowed to transition to T. Similarly, if the gate is being *pulled down*, then it is allowed to transition to F, and if there is a pending hazard, it is allowed to transition to X. The following definitions formalize the various sensitivities of a gate.

Definition 4.3.2. Let $\chi : V \longrightarrow \mathbb{B}$ and $g \in T_{\mathbb{B}}(V)$. $\chi(g)$ denotes the extension of χ to Boolean expressions.

Definition 4.3.3. Let $\chi : V \longrightarrow \mathbb{B}$ and let $\mathcal{O}_x$ be a gate defined such that

$$\mathcal{O}_x = \{g^+ \mapsto x\uparrow, g^- \mapsto x\downarrow\}.$$

- $A^{\uparrow}_{\chi}$ is a predicate on gates denoting that $\mathcal{O}_x$ is currently being *pulled up* with respect to χ, *i.e.,*

$$A^{\uparrow}_{\chi}(\mathcal{O}_x) \Leftrightarrow \chi(g^+) = \mathtt{T} \text{ and } \chi(g^-) = \mathtt{F}.$$

- $A^{\downarrow}_{\chi}$ is a predicate on gates denoting that $\mathcal{O}_x$ is currently being *pulled down* with respect to χ, *i.e.,*

$$A^{\downarrow}_{\chi}(\mathcal{O}_x) \Leftrightarrow \chi(g^+) = \mathtt{F} \text{ and } \chi(g^-) = \mathtt{T}.$$

- $A^{\updownarrow}_{\chi}(\mathcal{O}_x)$ is a predicate on gates denoting that $\mathcal{O}_x$ is *interfering*, or *shorted*, with respect to χ, *i.e.,*

$$A^{\updownarrow}_{\chi}(\mathcal{O}_x) \Leftrightarrow \chi(g^+) = \mathtt{T} \text{ and } \chi(g^-) = \mathtt{T}.$$

- $A^{\bullet}(\mathcal{O}_x)$ is a predicate on gates, denoting that $\mathcal{O}_x$ is being *invalidated* with respect to χ, *i.e.,*

$$A^{\bullet}_{\chi}(\mathcal{O}_x) \Leftrightarrow \chi(g^+) = \mathtt{X} \text{ or } \chi(g^-) = \mathtt{X}.$$

For the moment, ignore how the I set is computed and just assume that any pending hazard is contained in I. The semantics allows for the state of any operator output to either change or to stay the same. A state change from χ to a state χ' must satisfy the following property: *for all* $x \in V$ *such that* $\chi(x) \neq \chi'(x)$:

$$\chi'(x) = \mathtt{T} \Rightarrow A^{\uparrow}_{\chi}(\mathcal{O}_x)$$
$$\chi'(x) = \mathtt{F} \Rightarrow A^{\downarrow}_{\chi}(\mathcal{O}_x)$$
$$\chi'(x) = \mathtt{X} \Rightarrow x \in I.$$

As there are *many such* χ' in general, there are many possible *next states*; this is a reflection of the natural, per-gate concurrency that is expressed in the above constraints whenever $\chi'(x) \neq \chi(x)$.

There are two varieties of hazards that can occur in asynchronous circuits: interferences and instabilities. The first type of hazard, *interference*, occurs when a gate is being shorted; *e.g.,* the NAND gate of Figure 4.4, defined by production rules

$$\{\neg a_2 \vee \neg b_2 \mapsto x\uparrow, a_3 \wedge b_1 \mapsto x\downarrow\},$$

exhibits an interference when both guards evaluate to true, such as in a state where

$$\chi(a_2) = \mathtt{F}, \chi(a_3) = \mathtt{T}, \chi(b_1) = \mathtt{T}, \chi(b_2) = \mathtt{F}.$$

Definition 4.3.4. Let $\chi : V \longrightarrow \mathbb{B}$ and $\mathcal{O}_x$ a gate. $\mathcal{O}_x$ is *interfering* with respect to χ if and only if $A^{\updownarrow}_{\chi}(\mathcal{O}_x)$.

The second type of hazard is *unstable* behavior. This occurs when, at some state (χ, I), a gate $\mathcal{O}_x$ is enabled to transition (*i.e.,* there exists a legal execution step to a state (χ', I') where $\chi(x) \neq \chi'(x)$) but does not transition in the *actual* step to (χ', I') (*i.e.,* $\chi(x) = \chi'(x)$), *and* the inputs to $\mathcal{O}_x$ change when going from (χ, I) to (χ', I') in such a way that $\mathcal{O}_x$ is disabled from transitioning in the following step. This unstable behavior captures some of the non-atomic properties of gates in real circuits. If a gate begins to transition towards one rail, but is cut off before completing this transition, the output of the gate may be interpreted individually by subsequent transistors as either T, F, or as a non-Boolean value.

Taking the NAND again as an example, it is *enabled* in the state $(\chi, \emptyset)$ with

$$\chi(a_2) = \mathtt{T}, \chi(a_3) = \mathtt{T}, \chi(b_1) = \mathtt{T}, \chi(b_2) = \mathtt{T}, \chi(x) = \mathtt{T}$$

in that there exists a legal step $\chi'(x) \neq \chi(x)$, $\chi'(x) = \mathtt{F}$. However, suppose that instead of this transition happening, only the gate's *inputs* change, so that χ' is given by

$$\chi'(a_2) = \mathtt{F}, \chi'(a_3) = \mathtt{F}, \chi'(b_1) = \mathtt{T}, \chi'(b_2) = \mathtt{T}, \chi'(x) = \mathtt{T}.$$

This gate is no longer enabled in the sense that during the next step, say to (χ'', I''), x cannot transition to the other stable value, *i.e.,* $\chi''(x) = \mathtt{F}$ is impossible.

Definition 4.3.5. Let $\chi, \chi' : V \longrightarrow \mathbb{B}$ and $\mathcal{O}_x$ a gate. $\mathcal{O}_x$ is *unstable* with respect to χ, χ' if and only if

$$A^{\uparrow}_{\chi}(\mathcal{O}_x),\ \chi'(x) \neq \mathtt{T}, \text{ and } \neg A^{\uparrow}_{\chi'}(\mathcal{O}_x); \text{ or } A^{\downarrow}_{\chi}(\mathcal{O}_x),\ \chi'(x) \neq \mathtt{F}, \text{ and } \neg A^{\downarrow}_{\chi'}(\mathcal{O}_x).$$

The I set tracks all pending hazards, so that in a "step" from (χ, I) to (χ', I'), it must be ensured that I' contains (a) all *interferences* with respect to χ', as well as (b) all *instabilities* with respect to χ, χ'. In addition to these two hazard *origination* events, `X` values must also be allowed to propagate. This is formalized by creating two auxiliary sets I^+ and I^-. The I^+ set simply accumulates all of the new interferences and instabilities generated in going from χ to χ', and the I^- set includes all variables that have transitioned. The set $I \setminus I^-$ is then used to allow unresolved hazards to persist from I to I'.

Definition 4.3.6. Let $\chi, \chi' : V \longrightarrow \mathbb{B}$; the set of *new potential hazards* with respect to χ, χ', denoted $I^+_{\chi,\chi'}$ is defined such that

$$u \in I^+_{\chi,\chi'} \Leftrightarrow \mathcal{O}_u \text{ is unstable with respect to } \chi, \chi', \mathcal{O}_u \text{ is interfering with respect to } \chi', \text{ or } A^{\bullet}_{\chi'}(\mathcal{O}_u).$$

Similarly, the set of *non-persisting potential hazards*, denoted $I^-_{\chi,\chi'}$, is defined such that

$$u \in I^-_{\chi,\chi'} \Leftrightarrow \chi'(u) \neq \chi(u).$$

Definition 4.3.7. Let $\mathcal{P}$ be a proper PRS. The *computation step* relation, $\rightrightarrows$, is a binary relation on *execution*

states defined such that $(\chi, I) \rightrightarrows (\chi', I')$ if and only if for all $x \in V$ with $\chi(x) \neq \chi'(x)$:

$$\chi'(x) = \mathtt{T} \Rightarrow A^{\uparrow}_{\chi}(\mathcal{O}_x)$$
$$\chi'(x) = \mathtt{F} \Rightarrow A^{\downarrow}_{\chi}(\mathcal{O}_x)$$
$$\chi'(x) = \mathtt{X} \Rightarrow x \in I,$$
$$\text{and } I' = I^{+}_{\chi,\chi'} \cup I \setminus I^{-}_{\chi,\chi'}.$$

Definition 4.3.8. Let $\mathcal{P}$ be a proper PRS and $\vec{\sigma} = \langle \sigma_1, \sigma_2, \dots \rangle$ be an infinite sequence of states. $\vec{\sigma}$ is a *legal execution sequence*, if and only if for all $i \geq 1$, $\sigma_i \ \rightrightarrows \ \sigma_{i+1}$.

In what follows, computations are restricted so as to satisfy a few important sensibility requirements. Such a computation assumes (a) that the reset state is free of interferences, instabilities, and X values, and (b) that the reset state initializes forks with the same value on every branch. The restriction on fork branches simplifies several timing assumptions given in Section 4.4.

Definition 4.3.9. Let $\mathcal{P}$ be a proper PRS and $\vec{\sigma} = \langle \sigma_1, \sigma_2, \dots \rangle$ an execution sequence. σ_1 is called the *reset state*.

Definition 4.3.10. Let $\mathcal{P}$ be a proper PRS and $\vec{\sigma}$ an execution sequence with reset state $\sigma_1 = (\chi_1, I_1)$. $\vec{\sigma}$ is *proper* if:

- for all x, $\chi_1(x) \neq \mathtt{X}$; and $I_1 = \emptyset$; and
- for all gates $\mathcal{O}_x$, for all $y, y' \in x \longrightarrow \cdot$, $\chi_1(y) \ = \ \chi_1(y')$.

Lastly, it is useful to extend the notions of stability and non-interference beyond a single sequence.

Definition 4.3.11. Let $\mathcal{P}$ be a proper PRS and σ_1 a reset state. $\mathcal{P}, \sigma_1$ is stable and non-interfering if and only if all proper execution sequences, $\vec{\sigma}$, with reset state σ_1 are stable and non-interfering.

4.4 Timing

Reaching a timing closure for an asynchronous system tends to be considerably easier to achieve than for a similar synchronous design. Even so, as CMOS evolves and becomes ever more varied, and as entirely new paradigms are targeted, certain assumptions about timing become harder to satisfy [62]. This section gives formal meaning to the terms used to discuss common timing assumptions made for asynchronous circuits and then uses these terms to provide concrete definitions for DI, QDI, and SI systems.

4.4.1 Transition Causality

An important concept used to reason about the sequencing of transitions is the notion of *acknowledgment*. Acknowledgment embodies the causal relationship between the current inputs of an operator, say $\mathcal{O}_x$, and

a *transition* in the state of x, *e.g.,* from $\chi(x) = \texttt{T}$ to $\chi'(x) = \texttt{F}$. This chapter leverages the fact that all guard expressions of a proper PRS are in disjunctive normal form in order to say that each guard variable in a true-valued conjunctive clause is acknowledged when the target variable transitions.

Definition 4.4.1. Let $\vec{\sigma}$ be a proper execution sequence. Associate to $\vec{\sigma}$ an *acknowledgment relation*,

$$\leftrightarrows \subseteq V \times \mathbb{N} \times V,$$

and write $x \leftrightarrows_i x'$ when $(x, i, x') \in \leftrightarrows$. The relation is defined inductively such that

(a) $x \leftrightarrows_i x'$ if, letting

$$\mathcal{O}_{x'} = \{c_1 \vee \cdots \vee c_m \mapsto x' \uparrow, d_1 \vee \cdots \vee d_n \mapsto x' \downarrow\}$$

either

- $\chi_i(x') \neq \texttt{T}$, $\chi_{i+1}(x') = \texttt{T}$, and $\pi(x, c_j) > 0$ for some c_j such that $\chi_i(c_j) = \texttt{T}$; or
- $\chi_i(x') \neq \texttt{F}$, $\chi_{i+1}(x') = \texttt{F}$, and $\pi(x, d_j) > 0$ for some d_j such that $\chi_i(d_j) = \texttt{T}$;

(b) $x \leftrightarrows_i x'$ if $\mathcal{O}_{x'}$ is a wire with $x \longrightarrow x'$, and for some $x'' \in x \longrightarrow \cdot$, $x \leftrightarrows_i x''$; $\chi_{i+1}(x') = \chi_i(x)$; and letting j be the largest index less than i such that $y \leftrightarrows_j x$ for some y, $x \not\leftrightarrows_k x'$ for all $j < k < i$.

Condition (a) formalizes the well-known definition of acknowledgment as a causal relationship between transitions [60], and it extends the definition by allowing wires to acknowledge gates and gates to acknowledge wires. Condition (b) further extends acknowledgment to handle *inconsistencies* that can occur at certain forks. As an example, consider Figure 4.5. This figure completely exposes all forks from a segment of the circuit from Figure 4.1. Notice that gates and wires inherently "hold" state, so b is automatically *staticized*. Now, consider a proper execution sequence $\vec{\sigma}$, where σ_i is specified by Figure 4.5. In this state, the inverter $\mathcal{O}_a$ is enabled to transition, as are the wires $\mathcal{O}_{a_2}$, $\mathcal{O}_{a_3}$, $\mathcal{O}_{b_1}$, and $\mathcal{O}_{b_2}$. If the inverter output transitions but the wires do not, then $\chi_{i+1} = \chi_i[a \mapsto \texttt{T}]$, and by condition (a) of acknowledgment, $d_1 \leftrightarrows_i a$. Continuing with this example, suppose that the $\mathcal{O}_{a_1}$ wire transitions next, yielding $\chi_{i+2} = \chi_{i+1}[a_1 \mapsto \texttt{T}]$. By condition (a) of acknowledgment, $a \leftrightarrows_{i+1} a_1$, and by condition (b) $a \leftrightarrows_{i+1} a_2$ and $a \leftrightarrows_{i+1} a_3$. In some sense, O_{a_2} and O_{a_3} skipped a transition (legally), and condition (b) maintains a consistent notion of acknowledgment. Furthermore, acknowledgment "chains" give rise to a transitive version of acknowledgment.

Definition 4.4.2. Let $\vec{\sigma}$ be a proper execution sequence. Associate to $\vec{\sigma}$ a relation

$$\leftrightarrows^+ \subseteq V \times \mathbb{N} \times \mathbb{N} \times V,$$

and write $x \leftrightarrows^+_{[m,n]} x'$ when $(x, m, n, x') \in \leftrightarrows^+$. The relation is defined inductively such that

- if $x \leftrightarrows_i x'$ then $x \leftrightarrows^+_{[i,i+1]} x'$;
- if $x \leftrightarrows^+_{[m,n]} x'$ and $\chi_{n+1}(x') = \chi_n(x')$, then $x \leftrightarrows^+_{[m,n+1]} x'$;

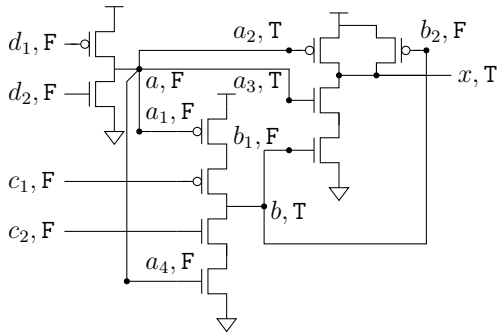


Figure 4.5: Simple buffer segment; at state $\sigma_i = (\chi_i, \emptyset)$.

- if $x \leftrightarrows^+_{[m,n]} x'$ and $x' \leftrightarrows_n x''$, then $x \leftrightarrows^+_{[m,n+1]} x''$.

4.4.2 Timing Assumptions

All of the timing assumptions presented in this chapter involve forks. Furthermore, these assumptions are defined and applied in terms of general n-way forks, as opposed to simply binary forks. The first such timing assumption is frequently overlooked because it places restrictions on forks internal to gates. These intra-operator forks are usually concealed by sharing variables across distinct conjunctive clauses within operator guard expressions, but they are made explicit by disallowing shared variables in every proper PRS. These forks are intentionally exposed, because they exist in real circuits, and they accurately account for a number of analog circuit constraints [30, 73].

The strong intra-operator fork assumption states that if any branch of a fork emanating from gate $\mathcal{O}_x$ has been acknowledged by a wire leading to another gate, say $\mathcal{O}_{x'}$, then all branches of the fork leading to $\mathcal{O}_{x'}$ have been acknowledged. This assumption is part of the standard gate-based digital circuit abstraction; *e.g.,* in CMOS circuits, it abstracts away details such as switching slew rates and relative transistor strengths. Additionally, it greatly simplifies the execution model, as hazard-free execution sequences are entirely within the *digital* realm; *i.e.,* at every step each variable can be interpreted as either $\mathtt{T}$ or $\mathtt{F}$.

Definition 4.4.3. Let $\vec{\sigma}$ be a proper execution sequence. $\vec{\sigma}$ satisfies the *strong intra-operator fork timing assumption* if and only if for all pairs of gates $\mathcal{O}_x, \mathcal{O}_{x'}$ and every index i;

$$\text{if } x \leftrightarrows_i y \text{ for some } y \in x \longrightarrow \cdot \longrightarrow x', \text{ then}$$
$$x \leftrightarrows_i y' \text{ for } \textit{all } y' \in x \longrightarrow \cdot \longrightarrow x'.$$

Consider a proper execution sequence $\vec{\sigma}$, where σ_i is specified by Figure 4.5, and the execution step where $\chi_{i+1} = \chi_i[a_2 \mapsto \mathtt{F}]$. This execution step does not satisfy the strong intra-operator fork timing assumption as $a \leftrightarrows_i a_2$ but not $a \leftrightarrows_i a_3$.

The next assumption is nearly identical but constrains forks branching out to distinct operators.

Definition 4.4.4. Let $\vec{\sigma}$ be a proper execution sequence. $\vec{\sigma}$ satisfies the *strong inter-operator fork timing assumption* if and only if for every gate $\mathcal{O}_x$ and index i

$$\text{if } x \leftrightarrows_i y, \text{ then for all } x' \text{ such that } x \longrightarrow \cdot \longrightarrow x' \neq \emptyset$$
$$x \leftrightarrows_i y' \text{ for } \textit{some } y' \in x \longrightarrow \cdot \longrightarrow x'.$$

Consider a proper execution sequence $\vec{\sigma}$, where σ_i is specified by Figure 4.5, since $\chi_i(a_1) \neq \chi_i(a_2)$, $\vec{\sigma}$ does not satisfy the strong inter-operator fork timing assumption.

Taken together, the strong intra-operator fork and inter-operator fork timing assumptions are equivalent to the standard isochronicity assumption.

Definition 4.4.5. Let $\vec{\sigma}$ be a proper execution sequence. $\vec{\sigma}$ satisfies the *strong fork timing assumption* (SFTA) if and only if it satisfies the properties of Definitions 4.4.3–4.4.4.

Defined next is the notion of an adversary path [58, 62], a specific type of acknowledgment path beginning at one branch of a fork and looping around to the target of another branch of the same fork.

Definition 4.4.6. Let $\mathcal{O}_x, \mathcal{O}_u, \mathcal{O}_v$ be distinct gates such that $x \longrightarrow \cdot \longrightarrow u \neq \emptyset$ and $x \longrightarrow \cdot \longrightarrow v \neq \emptyset$. In addition, let $y \leftrightarrows^+_{[h,k]} x$ for some $y \in \cdot \longrightarrow x$, such that for all m, $\chi_m(x) = \chi_{h+1}(x)$ with $h < m \leq k$. With respect to $y \leftrightarrows^+_{[h,k]} x$, an *adversary* is any acknowledgment path $x \leftrightarrows^+_{[i,j]} v \leftrightarrows^+_{[j,k]} x'$, with $i > h$ and $x' \longrightarrow \cdot \longrightarrow u \neq \emptyset$, and where for all $y'' \in x \longrightarrow \cdot \longrightarrow u$ and $h < l \leq k$, $\chi_l(y'') \neq \chi_i(x)$; all, such wires $\mathcal{O}_{y''}$ are referred to as isochronic branches of the fork.

Figure 4.6 completely exposes all inter-operator forks from Figure 4.1. For clarity, since the strong intra-operator fork timing assumption is assumed, intra-operator forks are not drawn. Consider a proper execution sequence $\vec{\sigma}$, where σ_i is specified by Figure 4.6. Now imagine that $\mathcal{O}_{a_1}$, the wire between the inverter and the C-element, transitions; *i.e.,* $\chi_{i+1} = \chi_i[a_1 \mapsto \mathtt{F}]$. Next, the C-element transitions, and $\chi_{i+2} = \chi_{i+1}[b \mapsto \mathtt{T}]$; there is now an acknowledgment path $a \leftrightarrows^+_{[i,i+1]} a_1 \leftrightarrows^+_{[i+1,i+2]} b$. This acknowledgment path is an adversary. Intuitively, this adversary path creates a potential instability at $\mathcal{O}_x$. For example, suppose that $\chi_{i+3} = \chi_{i+2}[b_1 \mapsto \mathtt{T}]$. This enables the NAND gate, $\mathcal{O}_x$, but the F at the output of the inverter, $\mathcal{O}_a$, can propagate to the a_2 input of the NAND gate at any step, disabling the NAND gate and generating an instability.

Definition 4.4.7. Let $\vec{\sigma}$ be a proper execution sequence. $\vec{\sigma}$ satisfies the *weak inter-operator fork timing assumption* if and only if $\vec{\sigma}$ contains no adversaries.

The weak inter-operator fork timing assumption yields a weaker assumption than SFTA. This weaker assumption is the *adversary path timing assumption* (APTA). Section 4.5 proves that the SFTA and APTA assumptions are equivalent with respect to the existence of hazards.

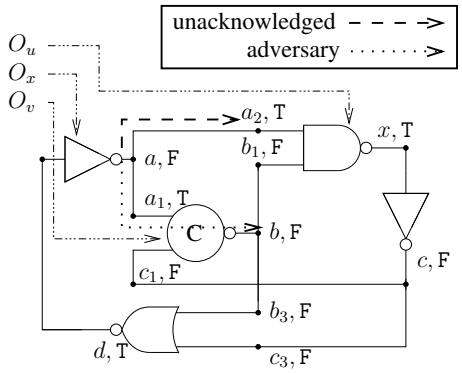


Figure 4.6: Adversary path; at state $\sigma_i = (\chi_i, \emptyset)$.

Definition 4.4.8. Let $\vec{\sigma}$ be a proper execution sequence. $\vec{\sigma}$ satisfies the *adversary path timing assumption* (APTA) if and only if it satisfies the properties of Definition 4.4.3 and Definition 4.4.7.

Next, following the classic definition [60], a circuit is defined as delay-insensitive if it is hazard-free under the assumption that wires, gates, and forks have arbitrary but finite delays.

Definition 4.4.9. Let $\mathcal{P}$ be a proper PRS. $\mathcal{P}$ is *delay-insensitive* (DI) with respect to reset state σ_1 if and only if for all proper execution sequences $\vec{\sigma}$ with reset state σ_1 and satisfying the properties of Definition 4.4.3, $\vec{\sigma}$ is stable and non-interfering.

Finally, this chapter provides a definition for quasi-delay-insensitive and speed-independent circuits. In agreement with [11], a circuit is SI if it is hazard-free under the assumption that gates and wires can have arbitrary delays, as long as these delays are positive and finite, but all wire forks must transition at the same time, *i.e.,* all sequences obey the strong intra-operator fork and strong inter-operator fork timing assumptions. Similarly, a circuit is QDI if it is hazard-free (stable and non-interfering) under the assumption that gates and wires can have arbitrary (positive and finite) delays with all sequences obeying the strong-intra operator fork timing assumption, and with a subset of the forks, called isochronic forks, additionally obeying the strong inter-operator fork timing assumptions.

Definition 4.4.10. Let $\mathcal{P}$ be a proper PRS. $\mathcal{F}$ denotes the subset of operators of $\mathcal{P}$ that are wires.

Definition 4.4.11. Let $\mathcal{P}$ be a proper PRS, let $\mathcal{F}_1, \mathcal{F}_2$ partition $\mathcal{F}$, and assume that the constraints of Definitions 4.4.3 and 4.4.4 are satisfied by all forks in $\mathcal{F}_1$, *i.e.,* they are isochronic, but that the forks in $\mathcal{F}_2$ are only required to satisfy the constraints of Definition 4.4.3.

(a) $\mathcal{P}$ is *speed-independent* (SI) w.r.t. to reset state σ_1 if and only if the set of proper execution sequences beginning with σ_1 are stable and non-interfering and $\mathcal{F}_2 = \emptyset$.

(b) $\mathcal{P}$ is *quasi-delay-insensitive* (QDI) w.r.t. to reset state σ_1 if and only if the set of proper execution sequences beginning with σ_1 are stable and non-interfering.

4.5 Equivalence of SFTA and APTA

Consider again Figure 4.5 and the operational definition of PRS computation from Section 4.3; there are a number of legal execution sequences from $\sigma_i = (\chi_i, \emptyset)$ that exhibit hazards. For example, there is an interference hazard when $\chi_{i+1} = \chi_i[b_1 \mapsto \mathtt{T}]$. The goal of timing assumptions, such as the strong intra-operator fork timing assumption (Definition 4.4.3), is to restrict the set of execution sequences so that hazards are excluded. Indeed, under the strong intra-operator fork timing assumption, the above execution step is impossible. Of course, different timing assumptions may exclude different execution sequences. Moreover, they may do so at different *costs*; *i.e.,* some timing assumptions may be *weaker* (easier to satisfy with physical circuits) than others.

This section primarily addresses the set of execution sequences that are excluded under (a) the *strong fork timing assumption* (SFTA, Definition 4.4.5), and (b) the *adversary path timing assumption* (APTA, Definition 4.4.8). The goal is to show that whenever the strong fork timing assumption excludes *all* execution sequences exhibiting a hazard, then so does the adversary path timing assumption; and *vice versa*. Formally, the aim is to prove the following theorem:

Theorem 4.5.1. *Let $\mathcal{P}$ be a proper* PRS *and σ_1 a reset state. $\mathcal{P}, \sigma_1$ is stable and non-interfering with respect to the strong fork timing assumption (*SFTA*) if and only if $\mathcal{P}, \sigma_1$ is stable and non-interfering with respect to the adversary path timing assumption (*APTA*).*

With respect to the interference hazard from Figure 4.5, when $\chi_{i+1} = \chi_i[b_1 \mapsto \mathtt{T}]$, both SFTA and APTA exclude the execution sequence, because both entail the strong intra-operator fork timing assumption. (It is worth noting that the strong intra-operator fork assumption may not be strictly necessary for proper SI circuit operation, but relaxing it falls clearly in the realm of analog constraints and is orthogonal to the equivalence of SFTA and APTA.) As a second example, modify the initial state of Figure 4.5 so that $\sigma_i = (\chi_i[d_1, d_2, a_1, a_4 \mapsto \mathtt{T}, b \mapsto \mathtt{F}], \emptyset)$. Consider the instability hazard exposed through the execution sequence $\chi_{i+1} = \chi_i[a_1, a_4 \mapsto \mathtt{F}]$, $\chi_{i+2} = \chi_{i+1}[b \mapsto \mathtt{T}]$, $\chi_{i+3} = \chi_{i+2}[b_1, b_2 \mapsto \mathtt{T}]$, and $\chi_{i+4} = \chi_{i+3}[a_2, a_3 \mapsto \mathtt{F}]$. $\mathcal{O}_x$ is enabled at index $i+3$, but no longer enabled at index $i+4$, so that $I_{i+4} = \{x\}$. Both timing assumptions again reject this sequence, but this time for different reasons. The execution step from i to $i+1$ is rejected under SFTA because $\chi_{i+1}(a_1) \neq \chi_{i+1}(a_2)$. Under APTA this execution step is allowed, but what is not allowed is the sequence of acknowledgments $a \leftrightarrows_i a_1 \leftrightarrows_{i+1} b$.

The ($\Leftarrow$) direction of Theorem 4.5.1 is straightforward, and is given in Section 4.5.1. The ($\Rightarrow$) direction is substantially harder. Section 4.5.2 sketches the main idea of the proof of Theorem 4.5.1 ($\Rightarrow$) at a high level. Sections 4.5.3 – 4.5.6 develop the details.

4.5.1 Theorem 4.5.1 ($\Leftarrow$)

Proof. Every SFTA execution sequence is also an APTA execution sequence. Toward a contradiction, assume there exists an execution sequence $\vec{\sigma}$ which is SFTA but not APTA. $\vec{\sigma}$ must contain an adversary path.

Let $x \ \leftrightarrows^{+}_{[i,j]} \ v \ \leftrightarrows^{+}_{[j,k]} \ y'$ be as in Definition 4.4.6. By the definition, for some index l, $i < l < j$, and all z, z' with $z \in x \longrightarrow \cdot \longrightarrow v$ and $z' \in x \longrightarrow \cdot \longrightarrow u$, $\chi_l(z) \neq \chi_l(z')$. This contradicts the strong inter-operator fork timing assumption. □

4.5.2 Theorem 4.5.1 ($\Rightarrow$) Overview

The proof of Theorem 4.5.1 follows by contradiction: assuming that SFTA is stable and non-interfering, it is shown that the existence of a hazardous APTA sequence implies also a hazardous SFTA execution sequence, an obvious contradiction. The proof given in the following sections is constructive, and so given an APTA execution sequence $\vec{\sigma}$ with a hazard, it is shown how to construct an SFTA execution sequence that also has a hazard.

The construction crucially relies on the notions of *relaxation* (Definition 4.5.2) and *variant* execution sequence (Definition 4.5.4). Given an APTA execution sequence, $\vec{\sigma}$, a variant is a *modified execution sequence*, say $\vec{\sigma}'$, in which certain transitions on wires are either *forced* or *suppressed*. In Figure 4.5, assuming again a modified initial state $\sigma_i = (\chi_i[d_1, d_2, a_1, a_4 \mapsto \mathtt{T}, b \mapsto \mathtt{F}], \emptyset)$, consider going from $\sigma_i = (\chi_i, \emptyset)$ to $\chi_{i+1} = \chi_i[a_1, a_4 \mapsto \mathtt{F}]$. This execution step is APTA but not SFTA because $\chi_{i+1}(a_1) \neq \chi_{i+1}(a_2)$. The condition where the branches of the $\mathcal{O}_a$ fork differ between gates $\mathcal{O}_x$ and $\mathcal{O}_b$ is called a *relaxation*, and the modifications that are made in a variant sequence are with respect to relaxations. One possible variant of the above execution step is to *force* a_2, a_3 to acknowledge a along with a_1, a_4, so that $\chi'_{i+1} = \chi_i[a_1, a_2, a_3, a_4 \mapsto \mathtt{F}]$; the second type of variant suppresses the acknowledgment of a on a_1, a_4, so that $\chi'_{i+1} = \chi_i$. Note that in both cases the modified execution sequence is SFTA.

A main insight of the proof is the identification of a gate, say $\mathcal{O}_u$, that is the *inherent origin of a hazard*. Moreover, the proof makes concrete the *forced/suppressed transitions needed to manifest this hazard at* $\mathcal{O}_u$. The gate is identified by considering the variant of $\vec{\sigma}$ in which *all relaxations are forced* (call this variant $\vec{\sigma}^{+}$) and is found at the smallest index, say j, where there is a gate, $\mathcal{O}_u$, such that $\chi_j(u) \neq \chi^{+}_j(u)$.

The details of Theorem 4.5.1 ($\Rightarrow$) are broken down as follows. Section 4.5.3 formally defines the notions of relaxation and variant. Section 4.5.3 also establishes (see Lemma 4.5.7) that *all variants are SFTA*. Section 4.5.4 isolates the hazard to a specific index and gate. Section 4.5.5 characterizes exactly how certain specific variants differ from the original APTA sequence. Finally, Section 4.5.6 demonstrates that the differences proved in the previous section are minor enough to yield a hazard in the SFTA variant when the APTA sequence has a hazard, which is finally proved in Section 4.5.6.

4.5.3 Relaxations and Variant Execution Sequences

The notion of *relaxation* encapsulates the idea that the first difference between the two timing assumptions manifests itself on the branches of a fork between two different gate operators. More specifically, using the weak inter-operator fork timing assumption, a signal may propagate to one gate at the end of a fork branch

and not to another gate at the end of a different branch, while, by definition, this is impossible under the strong inter-operator fork timing assumption.

Definition 4.5.2. Let $\vec{\sigma}$ be an APTA execution sequence. Associated to $\vec{\sigma}$ is a set of *relaxations*, $\mathcal{R}_{\vec{\sigma}}$, with

$$\mathcal{R}_{\vec{\sigma}} \subseteq V \times V \times \mathbb{N} \times \mathbb{N} \cup \{\infty\},$$

such that $(x, u, m, n) \in \mathcal{R}_{\vec{\sigma}}$ if and only if

- $\mathcal{O}_x, \mathcal{O}_u$ are gates, $m < n$, and $x \longrightarrow \cdot \longrightarrow u \neq \emptyset$.
- For some $y \in x \longrightarrow \cdot$; $x \leftrightarrows_m y$.
- For some $y \in x \longrightarrow \cdot$, $x \leftrightarrows_n y$; or $n = \infty$.
- For all $y \in x \longrightarrow \cdot$ and i such that $m < i < n$; $x \not\leftrightarrows_i y$.
- For all $y' \in x \longrightarrow \cdot \longrightarrow u$, $\chi_{m+1}(y') \neq \chi_m(x)$.

When $\vec{\sigma}$ is clear from context, $\mathcal{R}$ is used in place of $\mathcal{R}_{\vec{\sigma}}$. The essential idea behind constructing a *variant execution sequence* is to modify an APTA execution sequence *at relaxed forks*. There are two types of local modifications when a fork has a relaxation: the relaxed branches can be made to mimic the non-relaxed branches by *forcing transitions*; alternatively the non-relaxed branches can be made to mimic the relaxed branches by *suppressing transitions*. Conceptually, it is simpler to consider such modifications over a set of related relaxation points, a "*relaxation span*," rather than at every individual relaxation.

Definition 4.5.3. Let $\vec{\sigma}$ be an APTA execution sequence. The *relaxation span set*,

$$\mathcal{S}_{\vec{\sigma}} \subseteq V \times \mathbb{N} \times \mathbb{N} \cup \{\infty\}$$

is defined such that for every maximal sequence of relaxations

$$(x, u_1, m, i_1)\ (x, u_2, i_1, i_2)\ \cdots\ (x, u_k, i_{k-1}, n)$$

with $\chi_i(x) = \chi_m(x)$ for all $m \leq i \leq i_{k-1}$; $(x, m, n)\ \in\ \mathcal{S}_{\vec{\sigma}}$.

In the above definition, "maximal" means that there is no longer sequence, which includes the given one. Consider an (APTA) execution sequence $\vec{\sigma}$. The formal definition of variant attempts to mimic $\vec{\sigma}$ as closely as possible *except with the relaxation spans*. For a span $(x, m, n) \in \mathcal{S}$, the branches $y \in x \longrightarrow \cdot$ of the fork from $\mathcal{O}_x$ are either all *forced*, or all *suppressed*. Since every relaxation is part of a span, and across a span all branches of a fork are treated equally, a variant will always be SFTA. This is proved formally in Lemma 4.5.7.

The set $\mathcal{S}^+$ in the definition of variant corresponds to spans which are *forced*. The set $\mathcal{S}^-$ corresponds to

spans which are *suppressed*. The definition is broken up into pieces to facilitate explanation of the construction.

Definition 4.5.4. Let $\vec{\sigma}$ be an APTA execution sequence, and let $\mathcal{S}^+, \mathcal{S}^-$ partition $\mathcal{S}$. The *variant* of $\vec{\sigma}$ with respect to $\mathcal{S}^+, \mathcal{S}^-$ is the execution sequence, say $\vec{\sigma}'$, such that $\sigma'_1 = \sigma_1$, and ... (continued below)

For gates, $\vec{\sigma}'$ should *always* mimic $\vec{\sigma}$ if it can, and otherwise a default action should be taken. The default action forces the previous value of x to persist across the execution step.

Definition 4.5.5. Let $\mathcal{P}$ be a proper PRS and $\chi, \chi' : V \longrightarrow \mathbb{B}$. For any operator $\mathcal{O}_x$, χ, χ' *agree* on $\mathcal{O}_x$, $\chi(\mathcal{O}_x) \Leftrightarrow \chi'(\mathcal{O}_x)$, if and only if they give the same interpretation with respect to the predicates of Definition 4.3.3.

(**Definition 4.5.4 Cont., Gates).** ... for $i + 1 > 1$ and x such that $\mathcal{O}_x$ is a gate:

$$\chi'_{i+1}(x) = \chi_{i+1}(x) \qquad \text{if } \chi'_i(\mathcal{O}_x) \Leftrightarrow \chi_i(\mathcal{O}_x) \text{ and } y \leftrightarrows_i x \text{ for some } y \tag{1a}$$

$$\chi'_{i+1}(x) = \chi'_i(x) \qquad \text{otherwise} \tag{1b}$$

The same basic strategy employed for gates is also used for wires, *except* across a span.

(**Definition 4.5.4 Cont., Wires).** ... for $i + 1 > 1$ and y such that $\mathcal{O}_y$ is a wire with $y \in x \longrightarrow \cdot$:

$$\chi'_{i+1}(y) = \chi_m(x) \qquad \text{if } \chi'_m(\mathcal{O}_y) \Leftrightarrow \chi_m(\mathcal{O}_y) \text{ and there exists a } (x, m, n) \in \mathcal{S}^+ \text{ with } m \leq i < n \tag{2a}$$

$$\chi'_{i+1}(y) = \chi'_m(y) \qquad \text{if } \chi'_m(\mathcal{O}_y) \Leftrightarrow \chi_m(\mathcal{O}_y) \text{ and there exists a } (x, m, n) \in \mathcal{S}^- \text{ with } m \leq i < n \tag{2b}$$

$$\chi'_{i+1}(y) = \chi_i(x) \qquad \text{if } \chi'_i(\mathcal{O}_y) \Leftrightarrow \chi_i(\mathcal{O}_y), \text{ there is } \textit{no } (x, m, n) \in \mathcal{S}^+ \cup \mathcal{S}^- \text{ with } m \leq i < n, \text{ and } x \leftrightarrows_i y' \text{ for some } y' \in x \longrightarrow \cdot \tag{2c}$$

$$\chi'_{i+1}(y) = \chi'_i(y) \qquad \text{otherwise} \tag{2d}$$

It is straightforward to show that Definition 4.5.4 yields a well-defined execution sequence. Finally, Lemma 4.5.7 demonstrates that all variant execution sequences are SFTA.

Lemma 4.5.6. *Let $\vec{\sigma}$ be an* APTA *execution sequence and $\vec{\sigma}\,'$ a variant of $\vec{\sigma}$. $\vec{\sigma}\,'$ is a proper execution sequence with the same reset state as $\vec{\sigma}$.*

Proof. Sketch. $\vec{\sigma}$ and $\vec{\sigma}\,'$ have the same reset state by Definition 4.5.4. It can be shown that each state update given by Definition 4.5.4 is valid and hence; proof by induction follows directly from this. □

Lemma 4.5.7. *Let $\vec{\sigma}$ be an* APTA *execution sequence and $\vec{\sigma}\,'$ a variant of $\vec{\sigma}$. $\vec{\sigma}\,'$ is* SFTA.

Proof. By induction. It is sufficient to show that for all gates $\mathcal{O}_x$ and every index i, if $y, y' \in x \longrightarrow \cdot$, then $\chi'_i(y) = \chi'_i(y')$. At $i = 1$, $\sigma'_1 = \sigma_1$ and the result follows from the definition of a proper execution sequence reset state. Assume the result up to i; it must be shown to extend to $i + 1$.

Toward a contradiction, suppose $\chi'_{i+1}(y) \neq \chi'_{i+1}(y')$ for some such y, y' as above. It is easy to show that for any of the cases, if $\chi'_{i+1}(y)$ is defined by that case, then so is $\chi'_{i+1}(y')$. Clearly, both $\chi'_{i+1}(y)$ and $\chi'_{i+1}(y')$ cannot be defined by case (2a) or case (2c) (this would force $\chi'_{i+1}(y) = \chi'_{i+1}(y') = \chi_m(x)$ or $\chi'_{i+1}(y) = \chi'_{i+1}(y') = \chi_i(x)$, respectively); similarly, $\chi'_{i+1}(y), \chi'_{i+1}(y')$ cannot both be defined by case (2b) or both be defined by case (2d) (by the induction hypothesis). □

4.5.4 Isolating the Hazard

Let $\vec{\omega}$ be some unstable or interfering APTA execution sequence. This execution sequence is carried through the remainder of the proof of Theorem 4.5.1 ($\Rightarrow$), and is used to distinguish from $\vec{\sigma}$, which is used more generally. From $\vec{\omega}$ a "refined" APTA sequence is generated, $\vec{\omega}'$, and it is proved that a specific *variant* of $\vec{\omega}'$, $\vec{\omega}'^{-}$, also contains a hazard. This implies a contradiction because all variants are SFTA.

$\vec{\omega}'$ is constructed so as to isolate the hazard to an index j and specific gate $\mathcal{O}_u$. The construction is notable because the differences between $\vec{\omega}'$ and $\vec{\omega}'^{-}$ are extremely limited. The exact differences are given by Lemma 4.5.11. This allows for a relatively straightforward comparison of $\vec{\omega}'$ and $\vec{\omega}'^{-}$, showing that the variant sequence contains a hazard. Index j and gate $\mathcal{O}_u$ are found by comparing $\vec{\omega}'$ with $\vec{\omega}'^{+}$, the variant of $\vec{\omega}'$ where all relaxation spans are forced.

Definition 4.5.8. Let $\vec{\sigma}$ be an APTA execution sequence. $\vec{\sigma}^{+}$ denotes the variant of $\vec{\sigma}$ with respect to $\mathcal{S}, \emptyset$.

Refinement. Consider $\vec{\omega}$ and $\vec{\omega}^{+}$. Let j be the smallest index such that either $I_j \neq \emptyset$ or for some gate $\mathcal{O}_u$, $\chi_{j+1}(u) \neq \chi^{+}_{j+1}(u)$. Refine $\vec{\omega}$ to the execution sequence $\vec{\omega}'$ as follows ... (continued below)

The refinement branches based on the two conditions, *i.e.,* $I_j \neq \emptyset$ or not. The details of the first case are omitted and left as future work but are quite similar to when $I_j = \emptyset$. The second case is the key idea developed for the proof. $\chi_{j+1}(u) \neq \chi^{+}_{j+1}(u)$ indicates the potential for an instability hazard at $\mathcal{O}_u$. It will be shown that, essentially, if all of the relaxed forks leading to $\mathcal{O}_u$ at index j are forced to transition, then $\mathcal{O}_u$ becomes disabled. The remainder of the Refinement and proofs below deal with the details of demonstrating this result formally.

(**Refinement Cont.,** $(I_j = \emptyset)$)**.** Let $\mathcal{Z} \subseteq \mathcal{S}$ be such that for all $(x, m, n) \in \mathcal{S}$, $(x, m, n) \in \mathcal{Z}$ if and only if there is a $(x, u, k, l) \in \mathcal{R}$ with $m \leq k < j \leq l \leq n$. For all $i \leq j$ and $x \in V$

$$\chi'_i(x) = \chi_m(x) \qquad \text{if there exists a } (x, m, n) \in \mathcal{Z} \text{ with } m < i \leq n.$$
$$\chi'_i(x) = \chi_i(x) \qquad \text{otherwise}$$

for $i = j + 1$

$$\chi'_{j+1}(x) = \chi_m(z) \qquad \text{if there exists a } (z, m, n) \in \mathcal{Z} \text{ with } z \longrightarrow x.$$
$$\chi'_{j+1}(x) = \chi'_j(x) \qquad \text{otherwise}$$

and for all $i > j + 1$, $\omega'_i = \omega'_{j+1}$.

Lemma 4.5.9. *Let $\vec{\omega}$ and $\vec{\omega}'$ be as in the refinement; $\vec{\omega}'$ is an* APTA *execution sequence with the same reset state as w.*

Proof. Sketch. The main intuition as to why Lemma 4.5.9 is true comes from the fact that for $i \leq j$, $\vec{\omega}$ and $\vec{\omega}'$ differ at some gate $\mathcal{O}_x$ if and only if $(x, m, n) \in \mathcal{Z}$, where $m < j \leq n$. By Definitions 4.5.2 and 4.5.3, for all $y \in x \longrightarrow \cdot$, $m < k < n$, $x \not\rightleftarrows_k y$. That is, even if $\mathcal{O}_x$ transitions at some state, say ω_k, $m < k < n$, no wire could have observed this transition prior to ω_n. As such, x can clearly be held at its initial state in ω_m until ω_n. □

4.5.5 Constructing the Hazardous SFTA Sequence

The SFTA variant of $\vec{\omega}'$ that will be shown to contain a hazard is defined so that every relaxation span $(x, m, n) \in \mathcal{Z}$ is suppressed, while every other relaxation span is forced. This execution sequence is denoted $\vec{\omega}'^{-}$. Every difference between $\chi'_i(x)$ and $\chi'^{-}_i(x)$ is accounted for in Lemma 4.5.11 below. Unless x is the relaxed branch of a fork from a span $(x', m, n) \in \mathcal{S} \setminus \mathcal{Z}$, then the lemma essentially shows that $\chi'_i(x)$ is on an acknowledgment path from a suppressed $(x', m, n) \in \mathcal{Z}$.

Definition 4.5.10. Let $\vec{\sigma}$ be an APTA execution sequence, and $\mathcal{Z} \subseteq \mathcal{S}$. $\vec{\sigma}^{-}$ denotes the variant of $\vec{\sigma}$ with respect to $\mathcal{S} \setminus \mathcal{Z}$, $\mathcal{Z}$.

Lemma 4.5.11. *Let $\vec{\omega}'$, j, $\mathcal{O}_u$, and $\mathcal{Z}$ be as in the Refinement. With respect to $\vec{\omega}'^{-}$, for all $i \leq j$ and x, if $\chi'_i(x) \neq \chi'^{-}_i(x)$ then either*

(a) there is a $(x', m, n) \in \mathcal{S} \setminus \mathcal{Z}$ with $x \in x' \longrightarrow \cdot$ and relaxed at $m < i \leq n$, or

(b) there is a $(x', m, n) \in \mathcal{Z}$ such that $x' \leftrightarrows^{+}_{[m,i]} x$ in $\vec{\omega}'$.

Proof. Future work.

□

4.5.6 Theorem 4.5.1 ($\Rightarrow$)

Proof. ($I_j = \emptyset$)

Let $y \in \cdot \longrightarrow u$ be such that $\chi'^{-}_j(y) \neq \chi'_j(y)$. By Lemma 4.5.11, either (a) there is a $(x, m, n) \in \mathcal{S} \setminus \mathcal{Z}$ and $x \longrightarrow \cdot \longrightarrow u$ is relaxed at $m < j \leq n$, or (b) there is a $(x, m, n) \in \mathcal{Z}$ and $x \leftrightarrows^{+}_{[m,j]} y$ in $\vec{\omega}'$. Case (a) is impossible by the construction of the $\mathcal{Z}$ set, and case (b) is impossible by the definition of adversary path ($x \longrightarrow \cdot \longrightarrow u$ is relaxed at $m < j \leq n$, yet there is an acknowledgment from x at m that leads back to $\mathcal{O}_u$ at index j). Therefore, for all $y \in \cdot \longrightarrow u$, $\chi'^{-}_j(y) = \chi'_j(y)$. By similar reasoning, $\chi'^{-}_j(u) = \chi'_j(u) = \chi_j(u)$. It remains to be shown that $\chi'_{j+1}(y) = \chi^{+}_j(y)$ for all $y \in \cdot \longrightarrow u$ (this implies an instability in $\vec{\omega}'^{-}$). A simple corollary to Lemma 4.5.11 establishes that $\chi^{+}_j(y) \neq \chi'_j(y)$ exactly on those $y \in x \longrightarrow \cdot \longrightarrow u$ that are relaxed at j. By the construction of the refinement and case (2c) of the definition of variant, these are exactly the y that change in the execution step from χ'^{-}_j to χ'^{-}_{j+1}. Therefore, for all $y \in \cdot \longrightarrow u$, $\chi'^{-}_{j+1}(y) = \chi^{+}_j(y)$ and $\mathcal{O}_u$ gets disabled, an instability and a contradiction that SFTA is stable and non-interfering.

($I_j \neq \emptyset$) future work. □

4.6 Related Work

The importance of understanding timing assumptions in asynchronous circuits is well-known [60, 88]. Furthermore, it is recognized that strict *isochronicity* (SFTA) is both difficult to satisfy [93] and unnecessary for hazard-free operation. The adversary path is described directly in [62], and similar timing assumptions are described in [87]. These works provide important intuition as to why relaxing the strong fork timing assumption to the adversary path assumption is sufficient for ensuring hazard-free operation. However, they do so without a formal framework, and hence without proof of correctness. Moreover, they do not provide any intuition as to why the adversary path timing assumption is the *weakest* timing assumption that is both necessary and sufficient for correct operation of SI circuits.

Other works have generated useful extensions of SI circuits that relax isochronic forks, *e.g.,* the extended isochronic fork from [92]. The extended isochronic fork allows for additional gates to be placed on the unacknowledged branch of a fork and can yield more compact circuits. It seems clear that the adversary path assumption naturally extends to this assumption; although a formal proof establishing this is not given. Similarly, the timing constraint on orphans in NULL convention logic is almost certainly a specific variant of an adversary path; again, this is not formally established in this dissertation. However, by providing a

formalization at the level of switching networks, the current work could be extended to investigate such issues further.

4.7 Conclusion

This chapter presents a complete formalization of the notion of production rule sets, a well-known asynchronous computation system. Using this system, a number of fork-related timing assumptions are also formalized, including the adversary path assumption, and these formalizations are employed in order to characterize several important asynchronous logic frameworks. Finally, it is proved that the adversary path timing assumption is both a necessary and a sufficient condition for correct operation of speed-independent circuits and various extensions of SI circuits.

However, the model of computation presented, like all models, has limitations. First, it does not provide syntactic or semantic support for pass-transistors. Second, it does not directly include the transistors required to physically reset a PRS. Third, wires are assumed to be perfect. Finally, the model does not support interfering state-holding circuits such as cross-coupled inverters. The first two limitations are relatively minor; they have been excluded for clarity. The last two limitations require considerable effort to remedy without excessively encumbering the specification of the system, and are therefore left as future work.

Chapter 5

Real-World Application

During the course of my doctoral research I had the opportunity to design, build, tape out, and test a pair of radiation hardened ultra-low power QDI AVR microcontrollers: LP1 and DD1 (working as a team with Prof. Alain J. Martin and Chris Moore). I am deeply indebted to Alain for the opportunity and to Chris for performing more than his fair share of the engineering. Both LP1 and DD1 work to specification, and LP1 is able to operate reliably near the technology threshold voltage, consuming only 5.5pJ-per-instruction. As with any device, the choice of process technology plays a critical role in all aspects of the design, and in order to minimize power consumption (one of the primary goals of the project), I pushed our team to use a new cutting edge process (at the time): TSMC 40-nm low-power bulk CMOS (TSMC40LP). We had no design experience in this technology at the onset of the project, and many of the techniques developed throughout this dissertation were not mature enough for use in its evaluation. In this last technical chapter, I use the entirety of the work developed in my dissertation to analyze three critical design decisions—made with only partial knowledge early in the project—that were necessary in order to have high confidence in successful tape-out. This analysis was not possible prior to the development of the work presented in Chapters 2, 3, and 4.

5.1 Introduction

Two goals—minimize power and maximize reliability—drove the design and implementation of LP1. In order to achieve these goals and to maximize the likelihood of functional first-silicon, three conservative (and critical) design assumptions were made early in the project:

- the minimum energy operating point falls between 600mV and 700mV,
- if all adversary paths contain at least five gates in series, timing will not be violated, and
- by using combinational gates throughout (with the exception of SRAM) the chip will be robust.

The primary goal of this chapter is to apply the methods developed in this dissertation in order to analyze and quantify these assumptions in the TSMC40LP technology as applied to LP1. The secondary goal of

this chapter is to demonstrate how these methods of analysis can be easily and quickly applied to a real problem with minimal setup and computation cost. There may be some error in this back-of-the-envelope style of analysis, but the rigorous development of the models and careful quantification of error in Chapters 2 and 3, along with a formal proof about timing assumptions in Chapter 4, lend credence to an assumption of correctness. The organization of this chapter is as follows. In Section 5.1.1, the near-threshold model (developed in Chapter 2) is used to determine the minimum-energy operating point as a function of activity factor. Section 5.1.2 uses the near-threshold statistical delay model from Chapter 2 along with the adversary path timing assumption (see Chapter 4) to estimate the probability of a timing failure in LP1. Finally, the LP1-core robustness is estimated in Section 5.1.3 (see Chapter 3), and the probability of functional failure is compared to the probability of timing failure.

5.1.1 LP1 Minimum Energy Operating Point

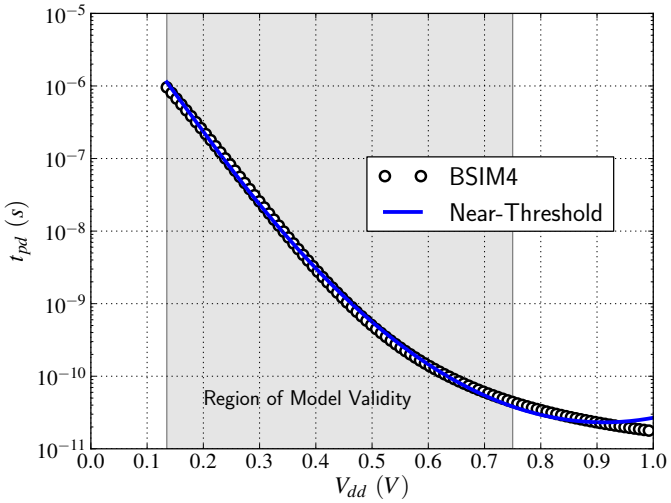


Figure 5.1: Inverter FO4 delay, Equation 2.44, plotted for entire V_{DD} range against a BSIM4 SPICE simulation of TSMC40LP ($25°\mathrm{C}, TT - Corner$) for minimum-size inverter driving an FO4 load. Fit from 135mV to 750mV, yielding $V_t = 515\mathrm{mV}$, $n = 1.60$, and $\frac{C_{\mathrm{load}}}{I_F} = 0.869\frac{ns}{V}$.

From the delay model derivation in Section 2.3.1 and from Equation 2.44, the FO4 delay of a minimum size inverter in TSMC40LP can be calculated and plotted as a function of V_{DD} (as depicted in Figure 5.1). Compared to BSIM4 SPICE simulation the mean absolute error is 18% and the maximum absolute error is 7.2%.

Using Equation 2.51, the leakage current (I_{off}), of minimum-size devices in TSMC40LP can be plotted and compared to BSIM4 SPICE simulations as depicted in Figure 5.2. The NFET I_{off} mean absolute error is 13% and the maximum absolute error is 4.1%; the PFET I_{off} mean absolute error is 16% and the maximum absolute error is 4.0%.

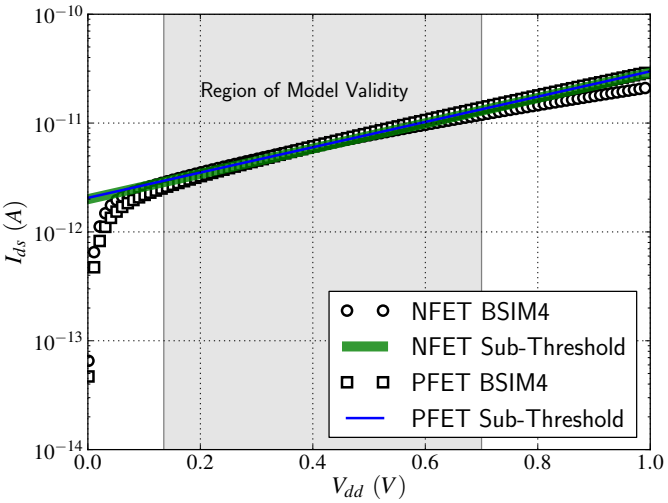


Figure 5.2: Off-current, Equation 2.51, plotted for entire V_{DD} range against a BSIM4 SPICE simulation of TSMC40LP (25°C, TT-Corner) for minimum-size devices with $V_t = 515$mV, $n = 1.60$. Fit from 135mV to 750mV, resulting in $\eta = 0.110$, NFET $I_0 = 552nA$, and PFET $I_0 = 190nA$.

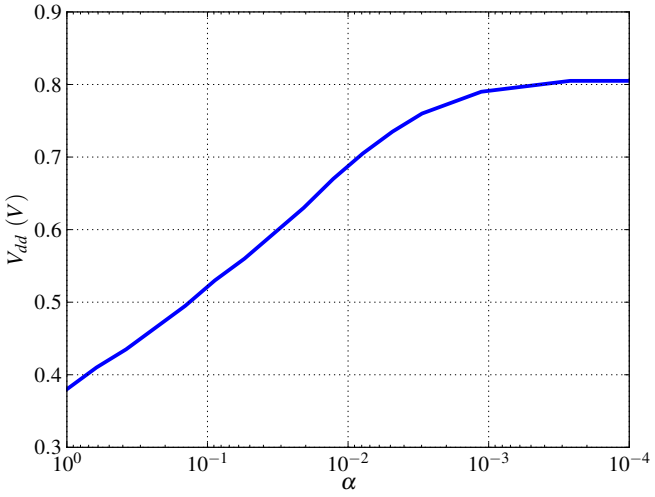


Figure 5.3: LP1 minimum-energy operating voltage vs. activity factor (α) (25°C, TT-Corner).

Finally, using the energy model derived in Section 2.3.2, the minimum energy operating point for the LP1 can be estimated. The LP1 operates at approximately 3,000 FO4 delays per cycle (this actually exceeds the performance requirements), so a corresponding path length of 3,000 FO4s is assumed; *i.e.,* $L_{dp} = N_l = 3,000$. In TSMC40LP the dynamic switching capacitance for an FO4 chain of inverters can be estimated as $C_{\text{dyn}} \approx 1.2fF * L_{dp}$. With this and the parameters taken from Figures 5.1 and 5.2, Figure 5.3 plots the estimated minimum-energy operating point for the LP1 as a function of activity factor. Minimum energy operation is achieved with a supply voltage ranging from approximately 400mV to 800mV depending on the activity factor.[1] The original target range of 600mV to 700mV falls well within this range but is too high to achieve minimum energy operation at high activities. Given that minimum energy operating range requires sub-threshold or near-threshold operation, reliability is a real concern; the probability of timing and noise margin failures must be analyzed and quantified.

5.1.2 LP1 Adversary Path Timing Failures

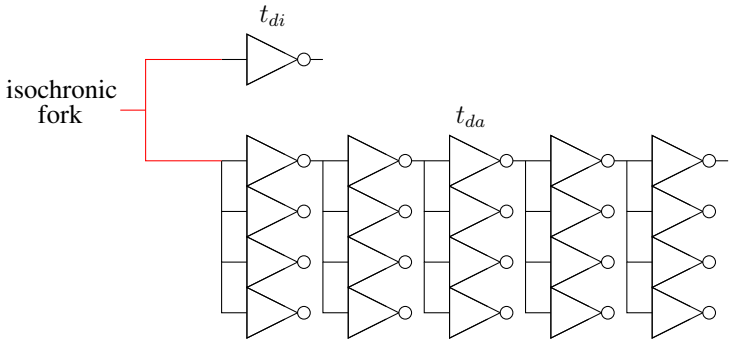


Figure 5.4: Depiction of the length-five simplified adversary path timing assumption. The delay on the isochronic branch is labeled as t_{di}, and the adversary path delay is labeled as t_{da}.

The rigorous definition and proofs from Chapter 4 can be reduced to a simple statement. *When designing QDI circuits using Martin Synthesis [58], only one type of timing assumption is made: every isochronic fork must satisfy the constraint that the isochronic branch of the fork transitions faster than the corresponding adversary path.* The LP1-core contains approximately 20K isochronic forks with adversary paths consisting of a variety of gates; however, there are at least five gates on every adversary path. Most of the LP1 adversary paths contain seven or more gates, and approximately 100 adversary paths contain only five gates in sequence. One simple approximation that provides an upper bound on the probability of failure is to consider an isochronic fork with a single FO1 inverter delay on the isochronic branch (t_{di}), and either five or seven FO4 inverter delays on the adversary path (t_{da}) as depicted in Figure 5.4; the assumption that $t_{di} < t_{da}$ is referred to throughout as the simplified adversary path timing assumption (SAPTA). It is reasonable to assume that the probability of an adversary path timing failure in the LP1 is strictly less than the probability

[1] The physical test data for the LP1 corroborates this, but is currently unpublished.

that either 100 instances of length-five SAPTA fail or 20K instances of length-seven SAPTA fail (*i.e.,* the probability of the union of these two events). The exact timing assumption is difficult to specify, because the transition on the isochronic branch does not cause a subsequent gate to switch. The isochronic branch of the fork must *tie* or *cut* a subsequent gate [61], thus preventing this gate from switching erroneously due to transitions on the adversary path. This chapter models this *tie* or *cut* time as a the shortest propagation delay, an FO1, and further justification and experimental confirmation is left as future work.

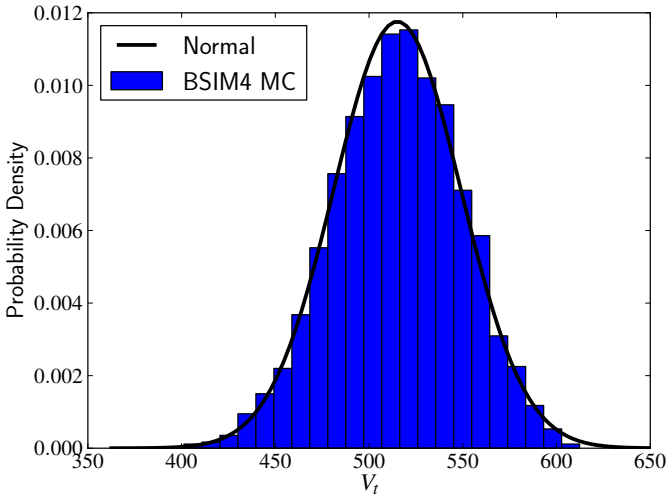


Figure 5.5: TSMC40LP NFET V_t distribution ($25°\mathrm{C}$, TT-Corner).

The TSMC40LP process is a low-power process with $V_t = 515\mathrm{mV}$ at $25°\mathrm{C}$ in the TT-Corner. As shown in Figure 5.5, V_t is a normally distributed RV with mean, $\mu_{V_t} = 515\mathrm{mV}$, and standard deviation, $\sigma_{V_t} = 34\mathrm{mV}$, (computed from statistical BSIM4 models using the methods from [33]). From Section 2.3.3, t_{di} and t_{da} can be modeled as log-normal RVs with expected values and variances given by Equation 2.60 and 2.61 respectively, where $L_{dp} = 1$ for t_{di} and $L_{dp} = 5$ *or* 7 for t_{da} (see Figure 5.6). Assuming independence, the probability that $t_{di} > t_{da}$ can be calculated by integration of the joint PDF. That is,

$$P(FAIL(SAPTA)) = P[t_{da} < t_{di}] = \int_{-\infty}^{\infty}\int_{-\infty}^{y} f_{t_{da}}(x) f_{t_{di}}(y) dx dy, \tag{5.1}$$

where $f_{t_{da}}(x)$ and $f_{t_{di}}(y)$ are the density functions for t_{di} and t_{da} respectively. For a log-normal RV the PDF follows from a change of variables on the normal PDF given in Equation 2.55. That is, If Z is a log-normally

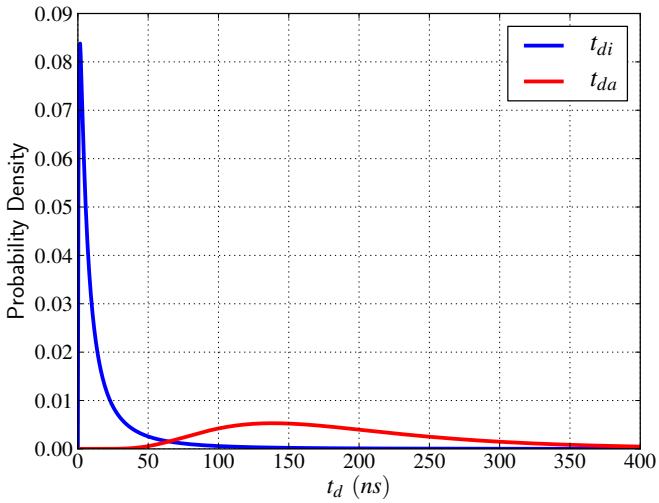


Figure 5.6: TSMC40LP t_{di} and length-five t_{da} distributions at $V_{DD} = 300$mV (25°C, TT-Corner). These log-normal PDFs are calculated from Equations 2.60 and 2.61, with parameters taken from Figures 5.1, 5.2, and 5.5. The small region of overlap (visible on the plot from 50ns to 100ns) makes clear the non-zero probability of an SAPTA timing failure.

distributed RV, then the corresponding probability density function (PDF), $f(Z)$, is given by

$$\begin{aligned} f(Z) &= \frac{1}{Z\sigma_Z\sqrt{2\pi}} e^{-\frac{(lnZ-\mu_Z)^2}{2\sigma_Z^2}}, \text{ where} \\ \mu_Z &= ln(E[Z]) - \frac{\sigma_Z^2}{2}, \text{ and} \\ \sigma_Z^2 &= ln\left(1 + \frac{VAR[Z]}{(E[Z])^2}\right). \end{aligned} \tag{5.2}$$

From Equations 2.58, 2.59, 2.60, and 2.61

$$E[t_{di}(V_t)] = \int_{-\infty}^{\infty} \frac{C_{\text{load}}}{I_F} \frac{V_{DD}}{\sigma_{V_t}\sqrt{2\pi}} e^{-k_1\frac{V_{DD}-V_t}{n\phi_t} - k_2\left(\frac{V_{DD}-V_t}{n\phi_t}\right)^2 - \frac{(V_t-\mu_{V_t})^2}{2\sigma_{V_t}^2}} dV_t, \tag{5.3}$$

$$Var[t_{di}(V_t)] = \int_{-\infty}^{\infty} \frac{C_{\text{load}}^2}{I_F^2} \frac{V_{DD}^2}{\sigma_{V_t}\sqrt{2\pi}} e^{-2k_1\frac{V_{DD}-V_t}{n\phi_t} - 2k_2\left(\frac{V_{DD}-V_t}{n\phi_t}\right)^2 - \frac{(V_t-\mu_{V_t})^2}{2\sigma_{V_t}^2}} dV_t - (E[t_{di}(V_t)])^2, \tag{5.4}$$

$$E[t_{da}(V_t)] = L_{dp}(da) \cdot E[t_{di}(V_t)], \text{ and} \tag{5.5}$$

$$Var[t_{da}(V_t)] = L_{dp}(da) \cdot Var[t_{di}(V_t)]. \tag{5.6}$$

As such, $P(FAIL(SAPTA))$ (Equation 5.1) can be directly computed by way of numerical integration. Figure 5.7 shows the estimated LP1 SAPTA failure probability vs. V_{DD}. Over the minimum energy oper-

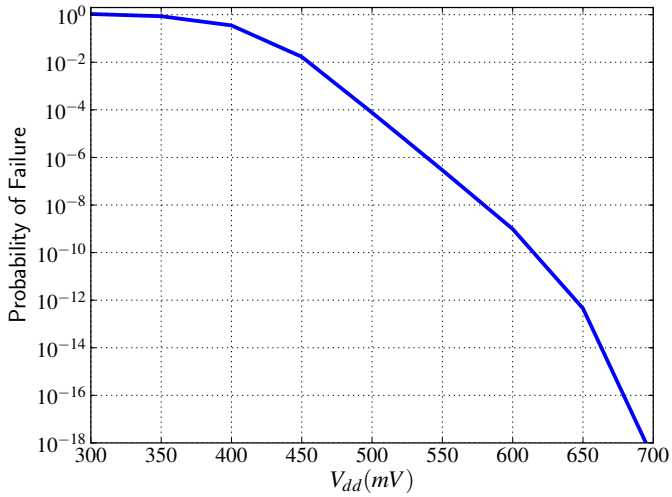


Figure 5.7: LP1 probability of isochronic-fork timing failures vs. V_{DD}. The probability is calculated by way of Equation 5.1, assuming 100 independent length-five and 20K length-seven SAPTA instances in TSMC40LP ($25°$C, TT-Corner). The parameters for Equation 5.1 are taken from Figures 5.1, 5.2, and 5.5.

ating range $\sim$ 400mV $-$ 800mV (see Figure 5.3); the LP1 timing failure probability ranges from 74% to astronomically unlikely. At 500mV (just slightly below the threshold voltage) the estimated probability of failure is approximately one in ten-thousand, or put another way, the expected yield loss is 0.01%—two to three orders of magnitude less than manufacturing defect yield loss. Of course, ensuring reliable timing is of little consequence if there is a high probability that gates simply fail to switch or are easily corrupted by noise.

5.1.3 LP1 Combinational Gate Robustness Estimate

It is possible to use the work developed in Chapter 3 to accurately compute the probability of SNM-based failures in LP1. At the time of the writing of this dissertation, the tools needed to perform this analysis are not complete, so this is left as future work. However, it is possible to derive a back-of-the-envelope robustness estimate. The LP1 contains full-custom radiation-hardened memories with a separate supply voltage from the core, so the memories are not considered in this analysis. The LP1-core also contains a few full-custom registers, but these are biased separately and also removed from this analysis. The LP1-core contains $\sim$16K standard cells. All of these cells (with the exception of the arbiter) are constructed using CMOS combinational logic, and the cell topology can be represented as $\sim$120K inverter equivalent pairs. Using the least robust gate pair (NAND3, NOR3) (as determined in Figure 3.31), an accurate upper bound on the probability of failure can be calculated. That is, for the purpose of calculating robustness, the LP1-core

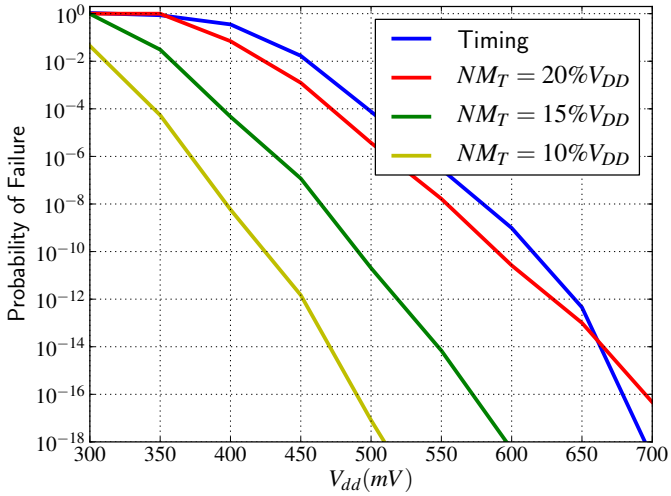


Figure 5.8: LP1 probability of isochronic-fork timing failures and robustness failures vs. V_{DD}. Timing failure probability is calculated by way of Equation 5.1, assuming 100 independent length-five and 20K length-seven SAPTA instances in TSMC40LP (25°C, TT-Corner). The parameters for Equation 5.1 are taken from Figures 5.1, 5.2, and 5.5. Robustness failures are calculated using Equation 3.26 with $\delta = -0.013$, and 120K (NAND3, NOR3) pairs in TSMC40LP (25°C, TT-Corner).

can be represented by chains of 120K minimum-size alternating NAND3 and NOR3 gates, and the probability of failure can be computed using Equation 3.26.

Figure 5.8 plots the LP1 robustness vs. V_{DD}, along with the probability of timing failure (from Figure 5.7). Noise-margin targets of $20\%V_{DD}$, $15\%V_{DD}$, and $10\%V_{DD}$ are depicted. Nonetheless—near the threshold voltage and to the first order—in a real QDI microcontroller core (LP1), the probability of a timing failure is greater than the probability of a noise-margin failure. Operation in the range of 300mV to 400mV comes at a significant risk of both timing and functional failures. In the energy-optimal range of 400mV to 800mV the failure probability quickly transitions from moderate (at 400mV) to negligible ($>$ 600mV).

Chapter 6

Conclusion and Future Work

6.1 Summary

This dissertation presents new models and methods for the analysis of minimum-energy QDI circuits. Chapter 2 details the physical derivation of the near-threshold model: a simplified and accurate transregional drain-current model for digital CMOS circuit analysis. This new model is compared with BSIM4 device models, and the error is reported. The near-threshold model is applied to several problems including determining the minimum energy operating point and modeling statistical path delay. Chapter 3 explores the effects of parameter variation on circuit robustness. A new metric to quantify statistical robustness is presented, and an efficient composable method of calculation is given. Chapter 4 formally defines the syntax and semantics of production rule sets (the object code used to build QDI circuits). This formal system is instrumental in the proof that properly designed and synthesized QDI circuits rely on the relatively easy-to-satisfy adversary path timing assumption. Finally, Chapter 5 uses the work developed in Chapters 2, 3, and 4 to analyze a QDI microcontroller developed at Caltech. The minimum energy operating point is determined, and the probabilities of timing failures and functional failures are calculated.

6.2 Discussion

The stated primary goal of this dissertation is to explore in detail the problems associated with building ultra-low-power QDI circuits in modern technologies. In [61], Martin discusses problems and solutions to building QDI circuits in high-variability technologies. In particular, he proposes a systematic method to replace *staticizers* (ratioed state-holding circuits) with combinational feedback, he suggests using longer adversary paths to mitigate timing failures, and he notes that QDI rings must consist of a sufficient number of gates in order to oscillate. These solutions improve reliability, but at a cost in power, performance, and area. This dissertation builds the analytical *tools* needed to reason about and to quantify these trade-offs, and then it demonstrates how to use them. Furthermore, these *tools* are not limited in application to QDI circuits; rather, they can be used to analyze any digital system.

In some sense, this dissertation attempts—by way of an extensive modeling effort—to provide insight into the nature of low-voltage digital circuits and systems in the face of parameter variation. In this context, QDI circuits serve an ancillary role; they provide a specific application for which to generate a general framework but are not the primary focus of the research. By intention, the near-threshold model (Chapter 2), the robustness metrics (Chapter 3), and the PRS syntax and semantics (Chapter 4) are entirely application agnostic. This broader-than-QDI-circuit applicability is essential, because the vast majority of digital circuits and systems in use today are synchronous. The reason behind this synchronous-circuit dominance is difficult to precisely determine. It may relate to the way in which the notion of computation is abstracted: the QDI abstraction is perhaps the most mathematically elegant, but the synchronous abstraction is probably simpler. On the other hand, it may be due to a practical engineering problem. QDI circuits currently cost approximately twice the area of their synchronous counterparts, but they should offer an increase in robustness at lower supply voltages [58]. This increase in robustness must be quantified in order to weigh the area/robustness/power trade-offs; perhaps the work presented within this dissertation can be used to better understand and quantify these trade-offs.

6.3 Future Work

Sections 2.5, 3.7, and 4.7 discuss several of the open problems that remain, and this section introduces a few other problems.

6.3.1 Near-Threshold Model

The near-threshold I_{on} model can be extended to act as a general I_{ds} model with which to analyze analog circuits, and include additional second-order digital effects (*e.g.,* the body effect). Generalizing the model comes at a real cost in complexity; in fact, it may result in an expression of similar complexity to that of EKV [37]. However, by using the Lambert W function and a more accurate approximation for inversion charge [78], it is possible to generate a new transregional I_{ds} model that is much more accurate than the EKV model at similar complexity. Such a model has the potential to aid analog circuit designers; in the analog realm, small improvements in model accuracy have a more significant impact on system design than in the digital realm.

6.3.2 Robustness

The robustness metric for chains of gates (proposed in Chapter 3) makes use of the static analysis of rings. Functional QDI circuits consist of interconnected oscillating rings; however, gates in oscillating rings spend most of a cycle attempting to hold state while waiting for a subsequent cycle, so a static analysis can serve as a first-order approximation. Nevertheless, a static analysis may not be sufficient to capture *every* failure

case; further investigation is necessary. An in-depth analysis of sub-threshold and near-threshold switching noise also remains largely open. It is likely that man-made noise in circuits operating sub-threshold and near-threshold is less (as a percentage of $\%V_{DD}$) than in the same circuits operating at the process nominal V_{DD} (the slew-rate degrades significantly as the supply is lowered).

Further validation of statistical circuit robustness may also prove fruitful. Verification of calculated robustness against statistical simulation of *real* circuits (*e.g.,* a 32-bit multiplier or a CPU) should serve to strengthen this work; however, the compute requirements are significant. A tiny transistor-level simulation of the LP1-core requires hundreds of core-hours of compute time using the most advanced fast-SPICE tools currently available. A meaningful statistical simulation requires millions of core-hours of computation. Regardless, further analysis of certain second-order effects (*e.g.,* correlation) is merited, and electronic design automation tools capable of handling large circuits should be completed. With such tools in place, new optimization algorithms to increase robustness—by way of sizing and gate selection—can be explored.

Lastly, it is possible to estimate circuit robustness using a simple ratio of device on-current to off-current, $\frac{I_{\text{on}}}{I_{\text{off}}}$. Alioto considers this problem in terms of an imbalance factor between PFET and NFET networks [4]. It may be possible to estimate the VTC parameters of a gate as a function of $\frac{I_{\text{on}}}{I_{\text{off}}}$, which in turn can be approximated in closed-form using the near-threshold and sub-threshold models.

Bibliography

[1] A. Agarwal, D Blaauw, V. Zolotov, and S. Vrudhula. Computation and refinement of statistical bounds on circuit delay. In *Design Automation Conference, 2003. Proceedings*, pages 348–353, 2003.

[2] A. Agarwal, K. Chopra, D Blaauw, and V. Zolotov. Circuit optimization using statistical static timing analysis. In *Design Automation Conference, 2005. Proceedings 42nd*, pages 321–324, 2005.

[3] M. Alioto. Understanding DC behavior of subthreshold CMOS logic through closed-form analysis. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 57(7):1597–1607, July 2010.

[4] M. Alioto. Ultra-low power VLSI circuit design demystified and explained: A tutorial. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 59(1):3–29, January 2012.

[5] D. B. Armstrong, A. D. Friedman, and P. R. Menon. Design of asynchronous circuits assuming unbounded gate delays. *Computers, IEEE Transactions on*, 18(12):1110–1120, 1969.

[6] P. Asenov, F. Adamu-Lema, S. Roy, C. Millar, A. Asenov, G. Roy, U. Kovac, and D. Reid. The effect of compact modelling strategy on SNM and read current variability in modern SRAM. In *Simulation of Semiconductor Processes and Devices (SISPAD), 2011 International Conference on*, pages 283–286, 2011.

[7] P. Asenov, N.A. Kamsani, D. Reid, C. Millar, S. Roy, and A. Asenov. Combining process and statistical variability in the evaluation of the effectiveness of corners in digital circuit parametric yield analysis. In *Solid-State Device Research Conference (ESSDERC), 2010 Proceedings of the European*, pages 130–133, 2010.

[8] Pieter Balk. 40 years MOS technology from empiricism to science. *Microelectronic Engineering*, 48(14):3–6, 1999.

[9] N.C. Beaulieu, A.A. Abu-Dayya, and P.J. McLane. Comparison of methods of computing lognormal sum distributions and outages for digital wireless applications. In *Communications, 1994. ICC '94, SUPERCOMM/ICC '94, Conference Record, 'Serving Humanity Through Communications.' IEEE International Conference on*, volume 3, pages 1270–1275, 1994.

[10] N.C. Beaulieu, A.A. Abu-Dayya, and P.J. McLane. Estimating the distribution of a sum of independent lognormal random variables. *Communications, IEEE Transactions on*, 43(12):2869–, 1995.

[11] P.A. Beerel, J.R. Burch, and T.H.-Y. Meng. Sufficient conditions for correct gate-level speed-independent circuits. In *Advanced Research in Asynchronous Circuits and Systems, 1994., Proceedings of the International Symposium on*, pages 33–43, 1994.

[12] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, and N. J. Rohrer. High-performance CMOS variability in the 65-nm regime and beyond. *IBM Journal of Research and Development*, 50(4.5):433–449, July 2006.

[13] Azeez J Bhavnagarwala, Xinghai Tang, and James D Meindl. The impact of intrinsic device fluctuations on CMOS SRAM cell stability. *Solid-State Circuits, IEEE Journal of*, 36(4):658–665, 2001.

[14] D. Bol, R. Ambroise, D. Flandre, and J. Legat. Analysis and minimization of practical energy in 45nm subthreshold logic circuits. In *Computer Design, 2008. ICCD 2008. IEEE International Conference on*, pages 294–300, 2008.

[15] D. Bol, R. Ambroise, D. Flandre, and J.-D. Legat. Interests and limitations of technology scaling for subthreshold logic. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 17(10):1508–1519, October 2009.

[16] S. Borkar, N.P. Jouppi, and P. Stenstrom. Microprocessors in the era of terascale integration. In *Design, Automation Test in Europe Conference Exhibition, 2007.*, pages 1–6, 2007.

[17] K.A. Bowman, B.L. Austin, J.C. Eble, Xinghai Tang, and J.D. Meindl. A physical alpha-power law MOSFET model. *Solid-State Circuits, IEEE Journal of*, 34(10):1410–1414, October 1999.

[18] B.H. Calhoun, Yu Cao, Xin Li, Ken Mai, L.T. Pileggi, R.A. Rutenbar, and Kenneth L. Shepard. Digital circuit design challenges and opportunities in the era of nanoscale cmos. *Proceedings of the IEEE*, 96(2):343–365, 2008.

[19] B.H. Calhoun and A. Chandrakasan. Analyzing static noise margin for sub-threshold SRAM in 65nm CMOS. In *Solid-State Circuits Conference, 2005. ESSCIRC 2005. Proceedings of the 31st European*, pages 363–366, 2005.

[20] B.H. Calhoun and A.P. Chandrakasan. Static noise margin variation for sub-threshold SRAM in 65-nm CMOS. *Solid-State Circuits, IEEE Journal of*, 41(7):1673–1679, July 2006.

[21] B.H. Calhoun, A. Wang, and A. Chandrakasan. Modeling and sizing for minimum energy operation in subthreshold circuits. *Solid-State Circuits, IEEE Journal of*, 40(9):1778–1786, September 2005.

[22] L. Chang, D.J. Frank, R.K. Montoye, S.J. Koester, B.L. Ji, P.W. Coteus, R.H. Dennard, and W. Haensch. Practical strategies for power-efficient computing technologies. *Proceedings of the IEEE*, 98(2):215–236, February 2010.

[23] Y.S. Chauhan and et al. Transitioning from BSIM4 to BSIM6. In *MOS-AK Workshop*, Delhi, India, March 2012.

[24] Jinhui Chen, L.T. Clark, and Yu Cao. Maximum - ultra-low voltage circuit design in the presence of variations. *Circuits and Devices Magazine, IEEE*, 21(6):12–20, 2005.

[25] Jinhui Chen, L.T. Clark, and Yu Cao. Robust design of high fan-in/out subthreshold circuits. In *Computer Design: VLSI in Computers and Processors, 2005. ICCD 2005. Proceedings. 2005 IEEE International Conference on*, pages 405–410, October 2005.

[26] B. Cheng, S. Roy, and A. Asenov. Impact of intrinsic parameter fluctuations on deca-nanometer circuits, and circuit modelling techniques. In *Mixed Design of Integrated Circuits and System, 2006. MIXDES 2006. Proceedings of the International Conference*, pages 117–121, 2006.

[27] B. Cheng, S. Roy, and A. Asenov. Impact of intrinsic parameter fluctuations on deca-nanometer circuits, and circuit modelling techniques. In *Mixed Design of Integrated Circuits and System, 2006. MIXDES 2006. Proceedings of the International Conference*, pages 117–121, June 2006.

[28] Sah Chih-Tang. Evolution of the MOS transistor-from conception to VLSI. *Proceedings of the IEEE*, 76(10):1280–1326, October 1988.

[29] Robert M. Corless, David J. Jeffrey, and Donald E. Knuth. A sequence of series for the Lambert W function. In *Proceedings of the 1997 international symposium on Symbolic and algebraic computation*, pages 197–204, New York, NY, USA, 1997. ACM.

[30] A. De Gloria, P. Faraboschi, and M. Olivieri. Design and characterization of a standard cell set for delay insensitive VLSI design. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 41(6):410–415, 1994.

[31] A. Devgan and C. Kashyap. Block-based static timing analysis with uncertainty. In *Computer Aided Design, 2003. ICCAD-2003. International Conference on*, pages 607–614, 2003.

[32] Li Ding and P. Mazumder. Dynamic noise margin: definitions and model. In *VLSI Design, 2004. Proceedings. 17th International Conference on*, pages 1001–1006, 2004.

[33] N. Drego, A. Chandrakasan, and D. Boning. Lack of spatial correlation in MOSFET threshold voltage variation and implications for voltage scaling. *Semiconductor Manufacturing, IEEE Transactions on*, 22(2):245–255, 2009.

[34] R.G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge. Near-threshold computing: Reclaiming Moore's law through energy efficient integrated circuits. *Proceedings of the IEEE*, 98(2):253–266, February 2010.

[35] C. Enz. An MOS transistor model for RF IC design valid in all regions of operation. *Microwave Theory and Techniques, IEEE Transactions on*, 50(1):342–359, January 2002.

[36] Christian C. Enz, Franois Krummenacher, and Eric A. Vittoz. An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications. *Analog Integrated Circuits and Signal Processing*, 8:83–114, 1995.

[37] Christian C. Enz and Eric A. Vittoz. *Charge-Based MOS Transistor Modeling: The EKV Model for Low-Power and RF IC Design*. Wiley, September 2006.

[38] S. Fisher, R. Dagan, S. Blonder, and A. Fish. An improved model for delay/energy estimation in near-threshold flip-flops. In *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pages 1065–1068, 2011.

[39] P. Friedberg, Yu Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos. Modeling within-die spatial correlation effects for process-design co-optimization. In *Quality of Electronic Design, 2005. ISQED 2005. Sixth International Symposium on*, pages 516–521, 2005.

[40] C. Galup-Montoro, M.C. Schneider, A.I.A. Cunha, F.R. de Sousa, H. Klimach, and O.F. Siebel. The advanced compact MOSFET (ACM) model for circuit analysis and design. In *Custom Integrated Circuits Conference, 2007. CICC '07. IEEE*, pages 519–526, September 2007.

[41] E. Grossar, M. Stucchi, K. Maex, and W. Dehaene. Read stability and write-ability analysis of SRAM cells for nanometer technologies. *Solid-State Circuits, IEEE Journal of*, 41(11):2577–2588, November 2006.

[42] D.M. Harris, B. Keller, J. Karl, and S. Keller. A transregional model for near-threshold circuits with application to minimum-energy operation. In *Microelectronics (ICM), 2010 International Conference on*, pages 64–67, December 2010.

[43] John R. Hauser. Noise margin criteria for digital logic circuits. *Education, IEEE Transactions on*, 36(4):363–368, 1993.

[44] C.F. Hawkins, J.M. Soden, A.W. Righter, and F.J. Ferguson. Defect classes-an overdue paradigm for CMOS IC testing. In *Test Conference, 1994. Proceedings., International*, pages 413–425, 1994.

[45] P. Heydari and M. Pedram. Ground bounce in digital VLSI circuits. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 11(2):180–193, 2003.

[46] C. F. Hill. Definitions of noise marign in logic systems. *Mullard Technical Communications*, 89:239–245, September 1967.

[47] C. F. Hill. Noise margin and noise immunity of logic circuits. *Microelectronics*, 1:16–21, April 1968.

[48] ITRS. *International Technology Roadmap for Semiconductors*, 2011.

[49] Michael Katelman, Sean Keller, and José Meseguer. Concurrent rewriting semantics and analysis of asynchronous digital circuits. In *Rewriting Logic and Its Applications*, volume 6381 of *Lecture Notes in Computer Science*, pages 140–156. Springer Berlin Heidelberg, 2010.

[50] Michael Katelman, Sean Keller, and José Meseguer. Rewriting semantics of production rule sets. *The Journal of Logic and Algebraic Programming*, 81:929–956, 2012. Rewriting Logic and its Applications.

[51] S. Keller, M. Katelman, and A.J. Martin. A necessary and sufficient timing assumption for speed-independent circuits. In *Asynchronous Circuits and Systems, 2009. ASYNC '09. 15th IEEE Symposium on*, pages 65–76, 2009.

[52] Sean Keller, Siddharth S. Bhargav, Chris Moore, and Alain J. Martin. Reliable minimum energy CMOS circuit design. In *2nd European Workshop on CMOS Variability (VARI 2011)*, Grenoble, France, May 2011.

[53] M.A. Korbel, D.C. Stow, C.R. Ferguson, and D.M. Harris. Yield-driven minimum energy CMOS cell design. In *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*, pages 1010–1014, 2012.

[54] J. Kwong and A.P. Chandrakasan. Variation-driven device sizing for minimum energy sub-threshold circuits. In *Low Power Electronics and Design, 2006. ISLPED'06. Proceedings of the 2006 International Symposium on*, pages 8–13, October 2006.

[55] A.W. Lo. Physical realization of digital logic circuits. In *Micropower Electronics*, pages 19–39. Macmillan, NY, 1964.

[56] J. Lohstroh, E. Seevinck, and J. de Groot. Worst-case static noise margin criteria for logic circuits and their mathematical equivalence. *Solid-State Circuits, IEEE Journal of*, 18(6):803–807, 1983.

[57] D. Markovic, C.C. Wang, L.P. Alarcon, Tsung-Te Liu, and J.M. Rabaey. Ultralow-power design in near-threshold region. *Proceedings of the IEEE*, 98(2):237 –252, February 2010.

[58] A.J. Martin and M. Nystrom. Asynchronous techniques for system-on-chip design. *Proceedings of the IEEE*, 94(6):1089–1120, 2006.

[59] Alain J. Martin. Compiling communicating processes into delay-insensitive VLSI circuits. *Distributed Computing*, 1(4):226–234, 1986.

[60] Alain J. Martin. The limitations to delay-insensitivity in asynchronous circuits. In *AUSCRYPT '90: Proceedings of the sixth MIT conference on Advanced research in VLSI*, pages 263–278. MIT Press, 1990.

[61] Alain J Martin. Asynchronous logic for high variability nano-CMOS. In *Electronics, Circuits, and Systems, 2009. ICECS 2009. 16th IEEE International Conference on*, pages 69–72, 2009.

[62] Alain J. Martin and Piyush Prakash. Asynchronous nano-electronics: Preliminary investigation. In *Asynchronous Circuits and Systems, 2008. ASYNC '08. 14th IEEE International Symposium on*, pages 58–68, 2008.

[63] J.D. Meindl and J.A. Davis. The fundamental limit on binary switching energy for terascale integration (TSI). *Solid-State Circuits, IEEE Journal of*, 35(10):1515–1516, October 2000.

[64] R E Miller. *Switching Theory, Volume II: Sequentical Circuits and Machines*. John Wiley & Sons, Inc., 1965.

[65] S. Mukhopadhyay, H. Mahmoodi, and K. Roy. Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 24(12):1859–1880, 2005.

[66] D E Muller, W Bartky, and S. A theory of asynchronous circuits. In *Laboratory of Harvard University, Vol. 29, Part I, Harvard University Press*, pages 204–243, 1959.

[67] M.H. Na, E.J. Nowak, W. Haensch, and J. Cai. The effective drive current in CMOS inverters. In *Electron Devices Meeting, 2002. IEDM '02. Digest. International*, pages 121–124, 2002.

[68] Koichi Nose and Takayasu Sakurai. Optimization of VDD and VTH for low-power and high speed applications. In *ASP-DAC '00: Proceedings of the 2000 Asia and South Pacific Design Automation Conference*, pages 469–474, New York, NY, USA, 2000. ACM.

[69] K. Okada and N. Onodera. Statistical modeling of device characteristics with systematic fluctuation. In *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, volume 2, pages 437–440, 2000.

[70] A. Ortiz-Conde, F.J. García Sánchez, and M. Guzmán. Exact analytical solution of channel surface potential as an explicit function of gate voltage in undoped-body MOSFETs using the Lambert W function and a threshold voltage definition therefrom. *Solid-State Electronics*, 47(11):2067–2074, 2003.

[71] Adelmo Ortiz-Conde, Francisco J García Sánchez, and Juan Muci. Exact analytical solutions of the forward non-ideal diode equation with series and shunt parasitic resistances. *Solid-State Electronics*, 44(10):1861–1864, 2000.

[72] H.C. Pao and C.T. Sah. Effects of diffusion current on characteristics of metal-oxide (insulator)-semiconductor transistors. *Solid-State Electronics*, 9(10):927–937, 1966.

[73] Karl Papadantonakis. Design rules for non-atomic implementation of PRS. Technical Report CaltechC-STR:2005.001, California Institute of Technology, 2005.

[74] Yu Pu, J.P. de Gyvez, H. Corporaal, and Yajun Ha. Statistical noise margin estimation for sub-threshold combinational circuits. In *Design Automation Conference, 2008. ASPDAC 2008. Asia and South Pacific*, pages 176–179, 2008.

[75] Jan Rabaey, Anantha Chandrakasan, and Borivoje Nikolic. *Digital Integrated Circuits, 2nd Ed.* Prentice Hall, 2003.

[76] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits. *Proceedings of the IEEE*, 91(2):305–327, February 2003.

[77] T. Sakurai and A.R. Newton. Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas. *Solid-State Circuits, IEEE Journal of*, 25(2):584–594, April 1990.

[78] Jean-Michel Sallese, Matthias Bucher, Franois Krummenacher, and Pierre Fazan. Inversion charge linearization in MOSFET modeling and rigorous derivation of the EKV compact model. *Solid-State Electronics*, 47(4):677–683, 2003.

[79] Jean-Michel Sallese, Matthias Bucher, and Christophe Lallement. Improved analytical modeling of polysilicon depletion in MOSFETs for circuit simulation. *Solid-State Electronics*, 44(6):905–912, 2000.

[80] A.J. Scholten, L.F. Tiemeijer, R. Van Langevelde, R.J. Havens, A.T.A. Zegers-van Duijnhoven, and V.C. Venezia. Noise modeling for RF CMOS circuit simulation. *Electron Devices, IEEE Transactions on*, 50(3):618–632, 2003.

[81] E. Seevinck, F.J. List, and J. Lohstroh. Static-noise margin analysis of MOS SRAM cells. *Solid-State Circuits, IEEE Journal of*, 22(5):748–754, 1987.

[82] Mingoo Seok, D. Sylvester, and D. Blaauw. Optimal technology selection for minimizing energy and variability in low voltage applications. In *Low Power Electronics and Design (ISLPED), 2008 ACM/IEEE International Symposium on*, pages 9–14, 2008.

[83] K.L. Shepard and V. Narayanan. Conquering noise in deep-submicron digital ICs. *Design Test of Computers, IEEE*, 15(1):51–62, Jan.-Mar. 1998.

[84] K.L. Shepard, V. Narayanan, and R. Rose. Harmony: static noise analysis of deep submicron digital integrated circuits. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 18(8):1132–1150, August 1999.

[85] B.J. Sheu, D.L. Scharfetter, P.-K. Ko, and M.-C. Jeng. BSIM: Berkeley short-channel IGFET model for MOS transistors. *Solid-State Circuits, IEEE Journal of*, 22(4):558–566, August 1987.

[86] A. Singhee and R.A. Rutenbar. Why quasi-Monte Carlo is better than Monte Carlo or latin hypercube sampling for statistical circuit analysis. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 29(11):1763–1776, November 2010.

[87] N. Sretasereekul and T. Nanya. Eliminating isochronic-fork constraints in quasi-delay-insensitive circuits. In *Design Automation Conference, 2001. Proceedings of the ASP-DAC 2001. Asia and South Pacific*, pages 437–442, 2001.

[88] K.S. Stevens, R. Ginosar, and S. Rotem. Relative timing [asynchronous design]. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 11(1):129–140, 2003.

[89] Dennis Sylvester, Kanak Agarwal, and Saumil Shah. Variability in nanometer CMOS: Impact, analysis, and minimization. *Integration, the VLSI Journal*, 41(3):319–339, 2008.

[90] R.R. Troutman. VLSI limitations from drain-induced barrier lowering. *Electron Devices, IEEE Transactions on*, 26(4):461–469, 1979.

[91] Yannis Tsividis and Colin McAndrew. *Operation and Modeling of the MOS Transistor, 3rd Ed.* Oxford University Press, USA, 2nd edition, 2011.

[92] K. van Berkel, F. Huberts, and A. Peeters. Stretching quasi delay insensitivity by means of extended isochronic forks. *Asynchronous Design Methodologies, 1995. Proceedings., Second Working Conference on*, pages 99–106, 1995.

[93] Kees van Berkel. Beware the isochronic fork. *Integration, The VLSI Journal*, 13(2):103–128, 1992.

[94] Darko Veberic. Having fun with Lambert W(x) function. *CoRR*, abs/1003.1628, 2010.

[95] N. Verma, J. Kwong, and A.P. Chandrakasan. Nanometer MOSFET variation in minimum energy subthreshold circuits. *Electron Devices, IEEE Transactions on*, 55(1):163–174, January 2008.

[96] Neil H. E. Weste and David Money Harris. *CMOS VLSI Design, 4th Ed.* Addison-Wesley, 2010.

[97] Bo Zhai, D. Blaauw, D. Sylvester, and K. Flautner. Theoretical and practical limits of dynamic voltage scaling. In *Design Automation Conference, 2004. Proceedings. 41st*, pages 868–873, 2004.

[98] Bo Zhai, D. Blaauw, D. Sylvester, and K. Flautner. The limit of dynamic voltage scaling and insomniac dynamic voltage scaling. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 13(11):1239–1252, November 2005.

[99] Bo Zhai, S. Hanson, D Blaauw, and D Sylvester. Analysis and mitigation of variability in subthreshold design. In *Low Power Electronics and Design, 2005. ISLPED '05. Proceedings of the 2005 International Symposium on*, pages 20–25, 2005.