MANAGING VARIABILITY IN VLSI CIRCUITS

by

Brian T. Cline

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering)
in The University of Michigan
2010

Doctoral Committee:

Professor David Blaauw, Chair
Professor Marios Papaefthymiou
Associate Professor Joanna Mirecki-Millunchick
Associate Professor Dennis M. Sylvester

DEDICATION

To my Family & Friends...

This dissertation is dedicated to my family and friends that supported, encouraged, and inspired me throughout my education. I'd especially like to thank my wife, Cecilia, my parents, Kevin and Barbara, and my sisters Kelly and Caitlin. I love you all.

ACKNOWLEDGEMENTS

Aside from the people that I dedicate my dissertation to, I would also like to offer my most sincere gratitude to a number of people. First, I would like to thank the following people for their support: my advisor, David Blaauw, Dennis Sylvester, and the rest of my dissertation committee: Professor Papaefthymiou and Professor Mirecki-Millunchick. I would also like to especially acknowledge my colleague Vivek Joshi for his collaboration on Chapter 4. I also express my deepest thanks to all of my colleagues that worked under Professors Blaauw and Sylvester during my years at the University of Michigan. Finally, I would like to acknowledge the Semiconductor Research Corporation, Freescale Semiconductor, and Mentor Graphics for their support throughout my doctoral studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Over the last two decades, Design for Manufacturing (DFM) has emerged as an essential field within the semiconductor industry. The main objective of DFM is to reduce and, if possible, eliminate variability in integrated circuits (ICs). Numerous techniques for managing variation have emerged throughout IC design: manufacturers design instruments with minute tolerances, process engineers calibrate and characterize a given process throughout its lifetime, and IC designers strive to model and characterize variability within their devices, libraries, and circuits. This dissertation focuses on the last of these three techniques and presents material relevant to managing variability within IC design. Since characterization and modeling are essential to the analysis and reduction of variation in modern-day designs, this dissertation begins by studying various correlation models used within Statistical Static Timing Analysis (SSTA). In the end, the study shows that using complex correlation models does not necessarily result in significant error reduction within SSTA, and that simple models (which only include die-to-die and random variation) can therefore be used to achieve similar accuracy with reduced overhead and run-time. Next, the variation models, themselves, are explored and a new critical dimension (CD) model is proposed which reduces standard deviation error in SSTA by ~3X. Finally, the focus changes from the timing analysis level and moves lower in the design hierarchy to the libraries and devices that comprise the backbone of IC

design. The final three chapters study mechanical stress enhancement and discuss how to fully exploit the layout dependencies of mechanically stressed silicon. The first of these three chapters presents an optimization scheme that uses the layout dependencies of stress in conjunction with dual-threshold-voltage (V_{th}) assignment to decrease leakage power consumption by ~24%. Next, the second of the three chapters proposes a new standard cell library design methodology, called "STEEL." STEEL provides average delay improvements of 11% over equivalent single-V_{th} implementations, while consuming 2.5X less leakage than the dual-V_{th} alternative. Finally, the stress enhanced studies (and this document) are concluded by a new optimization scheme that combines stress enhancement with gate length biasing to achieve 2.9X leakage power savings in IC designs without modifying V_{th}.

CHAPTER 1

INTRODUCTION

For the past forty years, the driving force behind the semiconductor industry has been device scaling and the ability to manufacture smaller geometries. Traditionally, in order to maintain electric fields of the same magnitude within these scaled devices, process engineers would also scale the device voltages (e.g., supply voltage and threshold voltage). Since the creation of the first microprocessors in the 1970's, supply voltage and gate length have decreased from ~15V and 10μm [1], respectively, to 0.9V and 32nm [2] in state-of-the-art technologies. In other words, over the past three decades geometries have scaled by ~1000X while supply voltage has only scaled by ~10X. This difference is partially illustrated in Figure 1.1, which shows supply voltage versus gate length over the last 25 years [3-4]. Voltage scaling has significantly lagged behind geometry scaling in modern-day technology nodes (starting around the 90nm process node in Figure 1.1) because process engineers can no longer scale the supply voltage, V_{DD}, or the threshold voltage, V_{th}, without significantly degrading reliability and exponentially increasing leakage power consumption. Consequently, devices manufactured in the latest technology nodes have higher effective electric fields than their predecessors. These increased electric fields can lead to a number of parasitic effects such as drain-induced barrier lowering (DIBL), gate-induced drain leakage (GIDL), mobility degradation, and hot carrier

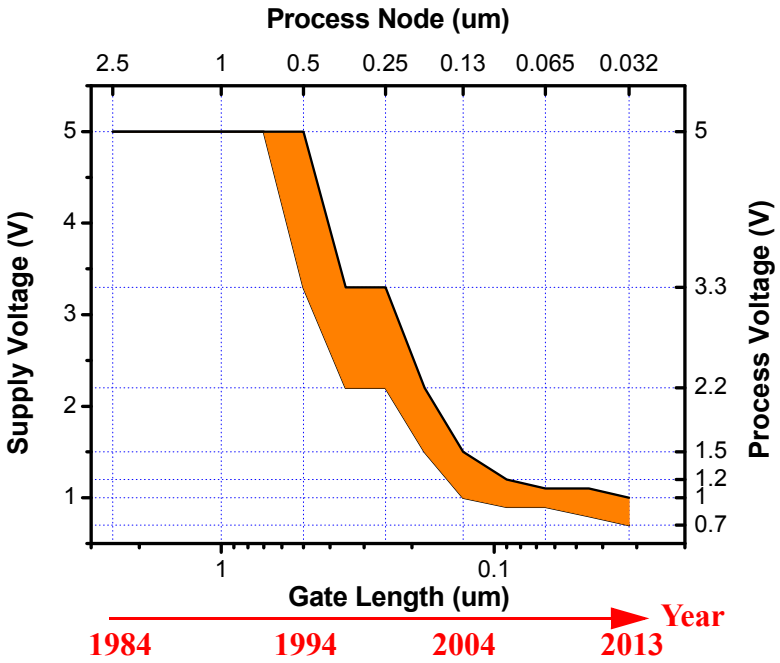


Figure 1.1. Supply Voltage vs. Process Node and Gate Length.
Voltage is shown as a range between high-performance and low-power voltages.

degradation. However, electric-field-related parasitics are merely one subset of a larger collection of issues that the semiconductor industry is faced with today.

Of all the pressing semiconductor issues, one of the most fundamental concerns is, simply, how to manufacture these nanoscale devices and fabricate features that are 32nm or smaller. Since state-of-the-art devices are currently made using photolithography techniques that use 193nm wavelength light, printing sub-193nm features on a wafer is difficult, due to optical effects that occur. To further complicate matters, manufacturing issues such as linewidth variation, random dopant fluctuation, and dielectric thickness variation have complex dependencies and are statistical in nature. This means that the traditional semiconductor device can no longer be handled in a deterministic manner and modern-day integrated circuits (ICs) have to be designed to tolerate variation in certain device parameters such as threshold voltage, gate length, and oxide thickness, in addition

to tolerating variation in interconnect properties like resistance and capacitance. In the last ten years variability in semiconductor devices has become such a large concern that an entirely new technology field has emerged – Design for Manufacturability (DFM).

The concept of designing with manufacturability in mind is somewhat of a departure from traditional semiconductor design practices since IC design and IC fabrication were two distinct entities for the first 30 years of the semiconductor industry. DFM, therefore, attempts to "bridge the gap" between these two fields and make engineers on both sides aware of the others' difficulties, challenges, and pitfalls. While linking IC design with semiconductor manufacturing, the ultimate objective is to improve IC yield by either reducing a circuit's susceptibility to variation or by reducing variation altogether. DFM from a "Very Large-Scale Integration" (VLSI) perspective typically involves reducing and tolerating certain amounts of variation in gate length (L), threshold voltage (V_{th}), oxide thickness (t_{ox}), and inter-layer dielectric (ILD) thickness.[1] Since the underlying mechanisms that cause variation in these parameters are different, each parameter requires its own set of solutions and design rules.

1.1 Gate Length Variation

Fabricated geometries in today's semiconductor processes vary from transistor to transistor, die to die, reticle to reticle, and wafer to wafer. Since digital ICs typically utilize the minimum gate length allowed for a device, gate length is especially susceptible to variation and can dramatically affect performance (in terms of both delay and power). Gate length variation is often included within a more liberal classification of variation,

[1] Inter-layer dielectric thickness is a measure of the dielectric height between metal layers in an IC.

called critical dimension (CD) variation.[2] CD variation has proven to be an interesting and difficult research problem on a variety of VLSI fronts. A significant number of publications have been dedicated to characterizing, modeling, analyzing, managing, and reducing CD variation [5-10]. CD variability is particularly formidable because it contains both a probabilistic component that is independent of other components, as well as a spatially correlated (systematic) component that is dependent on device context.[3] The probabilistic components of variation manifest themselves with either a low spatial frequency (e.g., shifts in CD) or high spatial frequency (e.g., line-edge roughness). The underlying causes of CD variation are numerous and include stepper imperfections (lens aberrations, variations in exposure and defocus, etc.), reticle defects, and photoresist variations (non-uniformity and thickness variation, post-exposure bake time variation, etc.), among others [7,11]. In fact, CD variability and its causes have become such a large concern that manufacturers have had to add mask correction techniques such as sub-resolution assist features (SRAFs) and optical proximity corrections (OPCs) to try and compensate for known imperfections during fabrication.

Process engineers rely on SRAF's and OPC's to ensure that the devices and interconnect print with minimum placement error (often referred to as EPE, or edge placement error). A simple example of what these features look like in a typical layout is included in Figure 1.2. In addition to improving printability, these features and corrections also strive to reduce variability. Other techniques that are being researched to reduce CD variability are regularity [8-9] and logic-brick/fabric design [10]. Since regularity makes

[2] Critical dimension refers to the smallest feature size that can be manufactured/printed in a particular technology.

[3] The context of a particular device involves both the distance between a device and its neighbors, as well as the size and orientation of the neighbors.

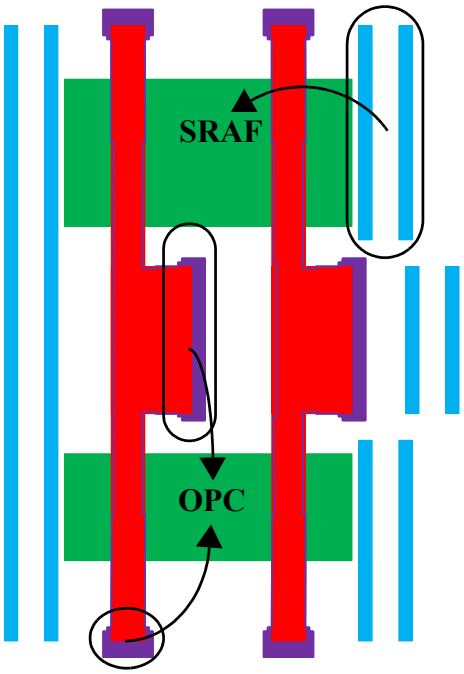


Figure 1.2. Simple Polysilicon SRAF and OPC Example.
Drawn poly is shown in red, while the OPC's and SRAF's are purple and blue.

context dependency more predictable (because features are placed at fixed intervals), it typically reduces the systematic CD variability. With reduced systematic variation, the complex OPC rules and resolution enhancement techniques (RETs) can be relaxed and become less computationally expensive.

Variations in CD affect VLSI designs in numerous ways and can dramatically alter an IC's performance. For example, gate length, or L, variation (one type of CD variation) affects a number of metal-oxide-semiconductor field-effect transistor (MOSFET) characteristics. Variation in L changes the drain current (I_D) in all operating regimes (subthreshold, triode, and saturation); the V_{th} through DIBL; and the gate-to-channel capacitance (C_{gc}), which loads the previous logic stage (modulating the previous stage's delay and dynamic power consumption). This means that for a given device, gate length variation will alter its propagation and rise/fall delays, its leakage power consumption, and

the delays and power consumption of its fanin cone.[4] Another example of CD variation is interconnect variation. Variation in the interconnect geometries modifies the capacitance and resistance of a given net. Variable interconnect capacitance affects both the coupling between nets, as well as the dynamic power consumption and delay of the gates driving those nets.

Given that CD variation affects so many circuit and device characteristics, accurately capturing this variability and developing techniques to handle it are essential to modern-day VLSI design. Typically in research, creating accurate models first involves characterizing the variability itself. In the case of CD variation, this requires capturing variations across dies, reticles, wafers, and lots. Additionally, since CD variation has a systematic component, it will also contain a certain amount of die-to-die, reticle-to-reticle, wafer-to-wafer, and lot-to-lot correlation. Characterizing this correlation and modeling it is another important aspect of capturing CD variability. Once the characteristics of CD variation are understood, accurate and efficient models can be extracted and used in timing analysis tools (discussed later in Section 1.5).

1.2 Threshold Voltage Variation

Another type of variation that impacts fundamental MOSFET device behavior is threshold voltage, or V_{th}, variation. The main cause of V_{th} variation is a purely probabilistic phenomenon (which is independent of other types of variation) known as random dopant fluctuation (RDF). Random dopant fluctuations occur in MOSFET devices because of the random nature of ion implantation [12-13]. However, with process scaling,

[4] The fanin cone of a net, *N*, is defined as the collection of gate(s) that have net *N* as an output.

the number of dopants located in a MOSFET's depletion region has decreased dramatically and is only on the order of hundreds in modern-day devices [14]. This fluctuation in channel dopants typically results in ~50mV of V_{th} variation in today's MOSFETs [14-15]. Similar to gate length and CD variation, threshold voltage variation has also been studied in detail and many people have proposed variation models [14-16]. On the other hand, V_{th} variation differs significantly from CD variation in that its main component is probabilistic and random in nature (aside from its dependency on gate length, itself). Therefore, V_{th} variation due to RDF is typically modeled as a Gaussian random variable that is characterized by its mean, μ, and standard deviation, σ [14-15].

Similar to CD, threshold voltage variation also influences a number of MOSFET device parameters. Both delay and leakage power are affected by changing V_{th} since drain current is dependent on threshold voltage. Delay is usually a linear or slightly super-linear function of V_{th} [17] while leakage power, on the other hand, is exponentially dependent on threshold voltage [18]. This exponential relationship between subthreshold current (and hence, leakage power) and V_{th} has become a major concern for contemporary VLSI designers. With billions of transistors in one design, leakage power consumption is now on the same order as dynamic power consumption (as illustrated in Figure 1.3), so any variation in leakage power can lead to significant variation in total circuit power. Additionally, V_{th} is often used as an optimization lever in VLSI circuits to achieve savings in either leakage power or delay [20-22]. However, with the amount of variability in V_{th} in sub-65nm devices, designers are becoming increasingly wary of using threshold voltage for optimization.

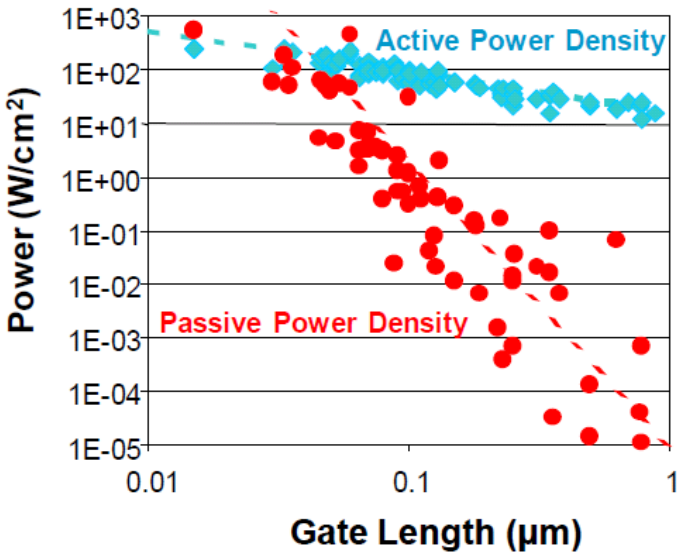


Figure 1.3. Dynamic and Static Power Density vs. Technology [19].

1.3 Oxide and Inter-Layer Dielectric Thickness Variation

In state-of-the-art process nodes, the equivalent gate oxide thickness, t_{ox}, is on the order of 1nm [23]. To put this in perspective, the silicon atom is ~0.2nm in diameter, which means that sub-65nm transistors have a gate oxide thickness that is *less than five silicon atoms thick*. Thus, atomic scale roughness introduced at the gate-to-oxide and oxide-to-silicon interfaces can cause significant amounts of oxide thickness variation (OTV) [24]. These variations are probabilistic in nature and can lead to variability in mobility, gate tunneling leakage current, and threshold voltage, among other parameters [24].

Aside from the gate oxide in today's devices, another type of dielectric material that experiences thickness variation is the dielectric between each metal layer in a process's metal stack. This material is often referred to as the inter-layer dielectric, or ILD. Inter-layer dielectric thickness variation is a spatially correlated (systematic) variation that is created during the Chemical-Mechanical Polishing (CMP) manufacturing step used to

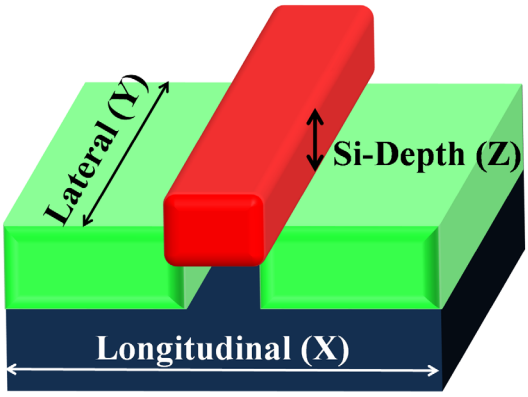
planarize dielectric material. With CMP, the resulting ILD thickness is dependent on topology because regions with higher interconnect density polish slower than sparse regions. Therefore, ILD thicknesses are spatially correlated with interconnect density and the variation can be predicted [25]. Due to this fact, numerous publications have provided techniques to improve metal density uniformity and, therefore, reduce the systematic variation in ILD due to CMP [26-27].

1.4 Sub-100nm Induced Variation and Mobility Degradation

As the semiconductor industry continues to scale below 100nm and approaches the fundamental limits of a number of parameters (e.g., CD size using 193nm wavelength light, t_{ox}, V_{th}, V_{dd}, etc.), process engineering becomes increasingly complicated. Effects like well proximity and mechanical stress due to shallow trench isolation (STI) have emerged in the last decade and now contribute to device variability. Furthermore, with the decline of voltage scaling, higher effective fields are causing increasing amounts of device parameter degradation due to phenomena like hot carriers and impact ionization. In recent process nodes, the amount of mobility degradation (due to the higher effective fields) has become so high that it has motivated the semiconductor industry to explore techniques like mobility enhancement. Currently, mobility enhancement is typically achieved by adding manufacturing steps to the process which induce mechanical stress in all MOSFET channels [28-31]. In the last five years, mechanical-stress-based enhancement has rapidly emerged across the semiconductor industry and many companies are employing one or more stress-enhancement techniques in their processes [28-32]. These techniques typically involve mechanical stress sources such as embedded-SiGe (in PMOS

source/drain regions) [28-29,31]; compressive/tensile (dual) nitride liners [28-30]; the Stress Memorization Technique (in NMOS transistors) [30]; and PMOS/NMOS hybrid orientation [32]. By inducing the correct type of stress in a MOSFET device, as shown in Figure 1.4, the effective mass and band scattering rates of the valence and conduction bands can be modified. These changes in effective mass and band scattering can result in increased mobility, which enhances transistor performance but increases leakage current.

While stress-based mobility enhancement does reduce the performance loss due to mobility degradation, it can also be a source of variation in today's devices because the sources of mechanical stress depend on layout properties like length of diffusion (LOD), contact placement, STI width, and well proximity [33-34]. In fact, experimental results show that MOSFET saturation current can vary by as much as 15% if stress dependencies are ignored. To date, researchers have mainly taken two different approaches regarding the layout dependency of stress: they either attempt to eliminate the dependency (using manufacturing techniques) [28,35] or they attempt to exploit the dependency (using layout techniques) [36].



	NMOS	PMOS
X	Tensile	Compressive
Y	Tensile	Tensile
Z	Compressive	Tensile

Figure 1.4. Preferred CMOS Device Stress Types.

1.5 Managing Variability in VLSI Designs

While process engineers constantly strive to mitigate the sources of variability discussed in the previous sections, the reality is that these sources are inherent to modern-day semiconductor manufacturing, so an intrinsic amount of variability is always present in manufactured IC's. Like any semi-automated manufacturing process, semiconductor manufacturing relies heavily on a number of different tools and instruments, and most of these instruments have to be calibrated frequently throughout the lifetime of a process in order to meet tolerance specifications for given parameters. It is the imperfections and non-zero tolerances of these tools that cause variability. Properties like stepper/scanner dosage and defocus, mask alignment, etch rate, etc., vary from wafer-lot to wafer-lot, wafer to wafer, reticle to reticle, and die to die. Since these tool imperfections typically affect specific stages of the manufacturing process, they are usually identified and classified by their region of impact: lot-to-lot, wafer-to-wafer, reticle-to-reticle, die-to-die, or within-die.

Furthermore, the semiconductor industry is continuing to scale device parameters, but the statistical mean of these parameters is decreasing more rapidly than the standard deviation (due to the intrinsic, probabilistic sources that cannot be eliminated or reduced). This means that the variation of a particular parameter is actually increasing with respect to its mean. Thus, over the last 10 to 20 years, IC designers have had to develop various methods of analysis and characterization which allow them to capture and reduce the variability of their designs.

Since static timing analysis (STA) [37] became the dominant timing verification method in modern VLSI design, the first techniques for managing variability involved identifying design corners and using STA to characterize circuits across various combinations of supply voltage, temperature, and process variation. At this point in semiconductor history, design corners were generally very simple since global/inter-die variations (types of variation that occurred from die to die; e.g., lot-to-lot, wafer-to-wafer, reticle-to-reticle, and within-reticle die-to-die) were more prevalent than local variations (also called intra-die or within-die variations). Therefore, circuit variation could be adequately captured by running characterization at the nominal-, best-, and worst-case process corners. At each corner, process variation for all devices in a circuit would be grouped into one category (e.g., worst-case process variation) and the STA program would then analyze the circuit given that all of its devices (and their parameters) were affected uniformly by this variation. For instance, in the worst-case design corner all device gate lengths and threshold voltages would be increased to their maximum possible value (under process variation), and then STA would characterize the circuit and report the decrease in performance.

In the last 20 years, however, local variations have grown in importance and were identified in the early 21st century as the dominant component [38-39]. During this time, corner-based analysis was labeled as "pessimistic" since the likelihood that *all* devices within a die would *all* be best- or worst-case at the same time was very small. The initial solutions to this criticism were to either run more corners or perform thousands of Monte Carlo STA analyses to determine the actual path distributions. The increased complexity incurred by these solutions was unattractive to the VLSI design community and

consequently spawned an entirely new area of research that explored propagating *statistical* distributions through a circuit graph, rather than *deterministic* delay values. This type of analysis was quickly labeled "Statistical Static Timing Analysis", or SSTA, and researchers sought to obtain more accurate, statistical representations of circuit performance [38-42].

In its simplest form, SSTA represents path delay as a weighted function of independent components [38-42]. However, since path delay is dependent on a number of varying parameters (L, t_{ox}, and V_{th}), modeling path delay as a function of these parameters and determining the sensitivity of delay to changes in each of these parameters is an essential component of SSTA. Thus, SSTA research is not only composed of proposed algorithms and related improvements, but it also includes modeling studies on various device parameters. The models typically used within SSTA for L, t_{ox}, and V_{th} variation were briefly mentioned in Sections 1.1 through 1.3.

While SSTA, in theory, produces a more accurate representation of delay than STA and corner-based analysis, actual implementations of SSTA algorithms have not distanced themselves from STA-based techniques, due to the simplistic underlying models and the approximations involved (e.g., the approximation that the maximum of two Gaussian variables is also Gaussian). Thus, additional research and improvements in both the underlying statistical process variation models, as well as the algorithm itself are needed to warrant the replacement of current deterministic timing analysis (STA-based) flows with their statistical counterpart.

1.6 Contribution of Dissertation

This dissertation focuses on two topics that are essential to the Design for Manufacturing space of integrated circuit design: CD variation and mechanical stress in silicon. Capturing, analyzing, and modeling CD variation is an important but difficult problem, as alluded to in Section 1.1. CD variation is different from V_{th}, t_{ox}, and ILD variation because it contains *both* a systematic component that is spatially correlated, and a probabilistic (random) component that is independent of other components. Variations in the other three parameters (V_{th}, t_{ox}, and ILD) originate from sources that are *either* probabilistic *or* systematic. This makes capturing and modeling their variability more manageable and straightforward. The CD variation research included in this dissertation began by analyzing raw CD data and characterizing the variations seen (die-to-die, reticle-to-reticle, wafer-to-wafer, etc.). Next, we used the data to compare a number of CD correlation models that had been proposed over the last decade. Prior to this work, the correlation models were presented from a conceptual perspective, but the actual implementation and accuracy in manufactured designs were not discussed.

Once the tradeoffs between correlation models were understood, we studied CD variation modeling within Statistical Static Timing Analysis (SSTA). Present-day CD models for timing analysis are error-prone because they do not capture the underlying sources of CD variability accurately. In fact, the models prior to this work grouped all CD variation (from various optical sources) across an entire standard cell library into one variable, essentially masking important, context-dependent effects that occur between transistors in a standard cell library. The CD variability research culminated in a new SSTA model that was more accurate than its predecessors.

The final DFM topic discussed in this dissertation is mechanical-stress-based mobility enhancement and its impact on circuit design. In modern processes, gate width (W), L, V_{th}, and t_{ox} are no longer the only parameters that affect a device's drain current (which impacts both performance and power consumption). The materials that process engineers now use to enhance MOSFET channel stress have their own dependence on layout (as discussed in Section 1.4), which results in device mobility variation. In order to characterize this mobility variation, this document concludes with a study that simulated, analyzed, and modeled the layout dependence of mechanical stress in silicon. After understanding the properties of mechanical stress, we proposed a novel standard cell library methodology, as well as a new timing optimization framework that combined mechanical-stress-enhancement with gate length biasing to achieve leakage power savings.

1.7 Organization of Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 focuses on critical dimension variation. It begins by analyzing electrical linewidth measurement (ELM) data obtained from fabricated 0.13μm technology device structures. This ELM data is then used to characterize and validate a number of correlation models that have been proposed over the last decade to accurately capture CD variation. At the end of the chapter, the results show that the basic correlation models provide a simpler solution than the complex models (in terms of overhead and run-time) and only increase error by a few percent. A discussion on modeling CD in SSTA follows in Chapter 3, which concludes by proposing a new SSTA model that accurately captures CD variation and reduces the

average error in standard deviation by ~3X. In Chapter 4, the variability focus shifts from CD to mechanical stress. It begins with a general discussion on mechanical stress in silicon and culminates in a technique that uses stress-enhancement in conjunction with dual-V_{th} assignment to reduce leakage by ~24%. Chapter 5 continues the discussion on mechanical stress, but deviates from the work in Chapter 4 in that it proposes a novel standard cell library technique and methodology for exploiting stress enhancement. This library methodology is used to improve delay (on average) by 11% over equivalent single-V_{th} implementations, while consuming 2.5X less leakage than the dual-V_{th} alternative. In Chapter 6, the stress-enhancement study is completed and a new optimization scheme that combines stress-enhancement with gate length biasing is presented. Results show that the proposed approach (stress plus gate length biasing) can optimize a single-V_{th} circuit to consume 2.5X less leakage than the dual-V_{th} approach with an average delay increase of only ~4%. Finally, Chapter 7 concludes the dissertation with a summary of the DFM work and a brief discussion of future work.

CHAPTER 2

CD VARIATION ANALYSIS AND CORRELATION MODELING IN SSTA

Static timing analysis (STA) has become a key method in the performance verification of modern chip designs and is the primary technique that abstractly incorporates manufacturing variation into design. Recently, the shortcomings of STA have become apparent with its inability to efficiently include within-die (or intra-die) variation in process parameters such as gate length, oxide thickness, and doping levels. STA, in its most common form, is a case-based analysis: designers perform simulations given best-, nominal-, and worst-case conditions and all devices are assigned the same process parameter value. However, with continued process scaling past 65nm, within-die variation has become more prominent and exhibited considerable spatial correlation. Unlike inter-die variation, within-die variation tends to average out over the length of a circuit path, which reduces the variance of a circuit's delay distribution. On the other hand, the presence of significant intra-die delay variation in two converging paths increases the "maximum" (typically Clark-based) delay distribution variance. With a case-based STA analysis, it is therefore difficult to construct a guaranteed bound on the actual timing distribution of a circuit without being overly conservative.

To address this issue, “Statistical Static Timing Analysis” (SSTA) was developed and it has received considerable attention in the CAD research community in recent years [38-41]. SSTA models process parameters, such as gate length and doping concentration, as random variables and propagates these random variables through the circuit in topological fashion, analogous to the propagation in its deterministic counterpart (STA).

The first efforts in SSTA modeled all process parameter variations, as well as the propagated arrival times, as independent random variables [42]. This assumption significantly simplified the analysis but compromised accuracy. In [40,43], process parameter variations were still considered independent, but correlations between arrival times due to reconvergence in the circuit were accounted for. In the latest generation of SSTA tools [38-39,41], correlations between the process parameters of different gates in the circuit are also considered.

Of the device parameters discussed in Chapter 1, typically only gate length (or more generally, CD) and inter-layer dielectric (ILD) thickness exhibit spatial correlations. Specifically, CD variation exhibits both a die-to-die component (causing all CD in a die to vary by some common amount) and a within-die component (where devices with close proximity are more likely to have similar CD). While die-to-die correlations can be incorporated relatively easily by enumerating a small number of die conditions, the within-die (spatial) correlations increase the complexity of SSTA substantially. Accounting for these correlations requires both a model which expresses the correlations in an amenable form, as well as an accompanying SSTA algorithm that can operate on that model. Over the past decade, a number of spatial correlation models have been proposed [38-39]. The spatial correlation model proposed in [39] used a grid-based approach where

the process parameters of all gates that fell within the same grid square were assumed to be identical. The correlation between different grid squares was decomposed using "Principal Component Analysis" (PCA), and then modeled as a weighted sum of independent random variables (the principal components). A different grid-based model was developed in [38]. Here, the authors combined multiple grids with varying granularity in a tree-like fashion, where each grid square was assigned an independent random variable and each gate was associated with every grid square in which it resided. While the Quad-tree used a larger total number of random variables than the PCA approach (given the same grid granularity), less information was associated with each individual gate. One important item addressed in this chapter that was not included in [38] is a method for fitting the Quad-tree model to measured data.

In this chapter, critical dimension (CD) data obtained through electrical linewidth measurements (ELM's) of a 0.13μm test chip design is used to analyze the accuracy of a number of proposed SSTA correlation models. The test chip consists of 8 different test structures (various densities and orientations of polysilicon lines) repeated at 308 sites per field over 23 fields and 5 wafers for a total of 35,420 measurements [44]. The ELM data is used to study both the correlation characteristics of actual CD variation as well as the effectiveness of different SSTA correlation models.

The remainder of the chapter is divided as follows. Section 2.1 discusses the types of gate length variation while Section 2.2 explains the spatial correlation models in more detail. Next, Section 2.3 demonstrates our experimental data and the results obtained while characterizing the raw ELM CD data. Section 2.4 implements the spatial correlation

models using the ELM data, discusses the observed model accuracy and, finally, Section 2.5 provides a brief summary of the results.

2.1 Types of Gate Length Variation

Within the random component of gate length variation, we can further distinguish three types of variation: independent, die-to-die, and spatially correlated. For this section, all variables – ΔL_x – are assumed to be zero-mean, unit-variance random variables.

2.1.1 Independent

In this type of variation, each device in the design has process parameter variations that are independent from the variations in other devices. Independent variations can be modeled using independent random variables. If the gate lengths in a die are completely specified by independent variations, the length of gate *i* can be expressed as follows:

$$L_{g,i} = L_{nom,i} + \sigma_{ri}\Delta L_{rnd,i}, \tag{2–1}$$

where $L_{nom,i}$ is the nominal value of gate length for that gate, $\Delta L_{rnd,i}$ is the random device length variation for gate *i*, and σ_{ri} is the sensitivity of gate *i* to changes in $\Delta L_{rnd,i}$.

2.1.2 Die-to-Die

Die-to-die variation, on the other hand, describes variation that is common for all devices on a particular die. When only inter-die variation is considered, all gate lengths within a particular die become perfectly correlated. Therefore, the gate length of gate *i*, only considering die-to-die variation, can be expressed as:

$$L_{g,i} = L_{nom,i} + \sigma_{dd}\Delta L_{die-to-die}, \quad (2\text{–}2)$$

where $L_{nom,i}$ is the nominal value for gate i, $\Delta L_{die\text{-}to\text{-}die}$ is a single random variable that is applied to all gates in the circuit, and σ_{dd} is the global gate sensitivity to changes in $\Delta L_{die\text{-}to\text{-}die}$.

2.1.3 Spatially Correlated

The last type of variation that we consider is spatially correlated variation. Most process variation within a single die is spatially correlated, and generally, correlation decays as a function of the distance between two points. Generally, in statistical timing analysis, the desire is to express correlation using a weighted sum of independent random variables, as shown below,

$$L_{g,i} = L_{nom,i} + \alpha_1\Delta L_1 + \alpha_2\Delta L_2 + \alpha_3\Delta L_3 + \ldots, \quad (2\text{–}3)$$

where ΔL_k is the variation of the k^{th} component and α_k is the sensitivity of the gate length to changes in the k^{th} component. By maintaining this form throughout the timing analysis, correlation information between the arrival times can be maintained. The specific values of the sensitivities and the number of components will vary between the different correlation models, which are discussed in the following subsection.

2.2 Correlation Models

The five correlation models analyzed in our experiments are die-to-die (D2D), independent (also referred to as "random," which we will use for the remainder of this chapter), D2D + random, PCA, and Quad-tree.

2.2.1 D2D, Random, and D2D + Random

The equations used to express the length variation of a particular gate for the random and die-to-die cases were shown in (2–1) and (2–2). Therefore, the "D2D + random" variation is a combination of (2–1) and (2–2):

$$L_{g,i} = L_{nom,i} + \sigma_{dd}\Delta L_{die-to-die} + \sigma_{ri}\Delta L_{rnd,i}, \tag{2–4}$$

Once we understand the forms of these gate length equations, it is simple to develop sensitivity matrices, which are the input to our statistical timing tool. For instance, the sensitivity matrices for D2D, random, and D2D + random are shown in (2–5) as D, R, and DR, respectively.

$$R = \begin{bmatrix} \sigma_{r1} & 0 & 0 & \cdots & 0 \\ 0 & \sigma_{r2} & 0 & \cdots & 0 \\ 0 & 0 & \sigma_{r3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{rn} \end{bmatrix} \qquad D = \begin{bmatrix} \sigma_{dd} & \sigma_{dd} & \sigma_{dd} & \cdots & \sigma_{dd} \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$DR = \begin{bmatrix} \sigma_{dd} & \sigma_{dd} & \sigma_{dd} & \cdots & \sigma_{dd} \\ \sigma_{r1}' & 0 & 0 & \cdots & 0 \\ 0 & \sigma_{r2}' & 0 & \cdots & 0 \\ 0 & 0 & \sigma_{r3}' & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{rn}' \end{bmatrix} \tag{2–5}$$

where σ_{dd} is the standard deviation of only the die-to-die component, σ_{ri} is the standard deviation of the i^{th} random component, and σ_{ri}' is the standard deviation of the i^{th} random component when the die-to-die component has been removed.

2.2.2 PCA

The PCA model is a grid-based model (shown in Figure 2.1) that separates the die into n grids. Each grid is associated with a principal component, and all n principal components are independent, normal random variables with zero mean and unit variance. Because PCA deals with spatially correlated distributions, its gate length equation is based on (2–3). Thus, for some gate i, its length can be expressed as:

$$L_{g,\,i} = L_{nom,\,i} + \sum_{j} \alpha_{ij} \Delta L_j, \text{ where } \alpha_{ij} = \sigma_i v_{ij} \sqrt{\lambda_j}, \tag{2–6}$$

where ΔL_j is the j^{th} principal component and α_{ij} is calculated as stated in (2–6); σ_i is the standard deviation associated with grid i, v_{ij} is the i^{th} element in the j^{th} eigenvector of the correlation matrix, and λ_j is the j^{th} eigenvalue of the correlation matrix [39]. Therefore, the sensitivity matrix, P, for the PCA model will be of the form,

$$P = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \cdots & \alpha_{1,m} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \cdots & \alpha_{2,m} \\ \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} & \cdots & \alpha_{3,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{n,1} & \alpha_{n,2} & \alpha_{n,3} & \cdots & \alpha_{n,m} \end{bmatrix}, \tag{2–7}$$

where each grid is associated with one column and one row (and $m \le n$).

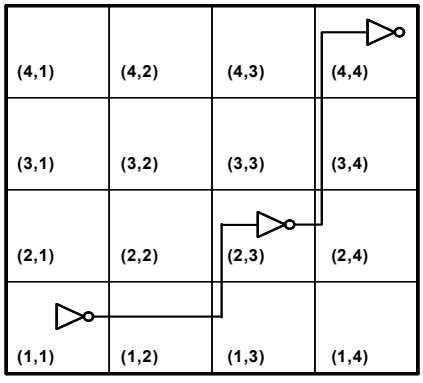


Figure 2.1. PCA Grid Example.

2.2.3 Quad-Tree

Quad-tree is another grid model that utilizes various grid levels combined in a tree-like structure – shown in Figure 2.2 – to include spatial correlation. The Quad-tree has l+1 levels, and each level, k, contains 2^k-by-2^k squares [38]. Levels are numbered where "level 0" represents the top level and l is bottommost level. Level 0 only has one grid, while level k has 4^k grids. All of the regions at different levels of the tree are associated with an independent random variable that includes part of the total intra-die variation. For a gate located within bottommost region r, the associated variation is a sum of all the intra-die variation components that intersect region r as you progress up the tree (e.g., in Figure 2.2 grid (2,13) intersects grids (1,3) and (0,1)). For example, the equation for gate length for the gate that lies in grid (2,7) is,

$$L_{g,(2,7)} = L_{nom,(2,7)} + \Delta L_{(2,7)} + \Delta L_{(1,2)} + \Delta L_{(0,1)}. \tag{2–8}$$

Thus, the sensitivity matrix is similar to the PCA matrix in (2–7), where all grids (including all levels of the tree) are given one row in the sensitivity matrix, but, in the

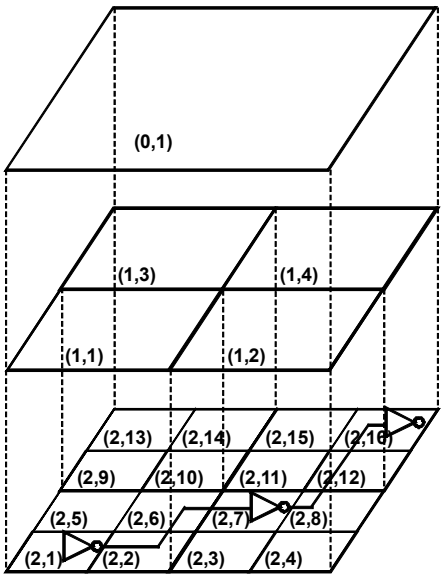


Figure 2.2. Quad-tree Model Example.

Quad-tree matrix, only the bottommost grids are assigned to individual columns. All grids that do not intersect with a particular bottommost grid (assigned column i in the matrix) will have a zero-sensitivity value at its row j in the matrix (i.e., element [j,i] equals zero). Equation (2–9) below contains the general form for a 3-level (l = 2) Quad-tree sensitivity matrix. Specific grids are shown in parentheses and there are a total of 16 (4^l=16) columns and 21 rows (1 "level 0" row + 4 "level 1" rows + 16 "level 2" rows). It is interesting to note that while this matrix has a larger number of elements compared to the equivalent PCA matrix ($16 \times 21 = 336$ compared to $16 \times 16 = 256$), the majority of the 336 elements are 0, making the Quad-tree version a sparse matrix.

$$Q = \begin{bmatrix}
\alpha_{(0,1)} & \alpha_{(0,1)} & \alpha_{(0,1)} & \alpha_{(0,1)} & \cdots & \alpha_{(0,1)} & \alpha_{(0,1)} & \alpha_{(0,1)} & \alpha_{(0,1)} \\
\alpha_{(1,1)} & \alpha_{(1,1)} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & \alpha_{(1,2)} & \alpha_{(1,2)} & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & \alpha_{(1,3)} & \alpha_{(1,3)} & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & \alpha_{(1,4)} & \alpha_{(1,4)} \\
\alpha_{(2,1)} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & \alpha_{(2,2)} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & \alpha_{(2,3)} & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \alpha_{(2,4)} & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & \alpha_{(2,13)} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & \alpha_{(2,14)} & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & \alpha_{(2,15)} & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \alpha_{(2,16)}
\end{bmatrix} \tag{2–9}$$

As stated in the introduction, the authors in [38] did not explain how to fit actual data to the Quad-tree model. After examining several different algorithms, we derived a Quad-tree fit that was efficient, simple and provided good accuracy. Prior to fitting, we

discovered that closely matching the die-to-die component was very important to the overall accuracy. Therefore, this fitting method was designed to accurately capture the die-to-die component first with zero error. The pseudo-code for the fitting algorithm is as follows:

Algorithm 2–1 QUADTREE_FIT(L) // Correlation fit for quad-tree
// L = number of levels

```
1: i_L = 0        // i_L = index of level in quad-tree
2: while (i_L < L)
3:    i_G = 1     // i_G = current grid number
4:    while (i_G < 4^(i_L))
5:       Compute grid mean, μ_i
6:       Computer the standard deviation, σ_μ,i , of μ_i for all dies
7:       Enter σ_μ,i into sensitivity matrix
8:    end while
9: end while
```

Essentially, the fitting method starts at level 0 and traverses down the tree. The method stops at each level, i_L, and determines the number of grids that comprise it (4^{i_L} for grid level i_L). Next, every grid on the particular level (all i_G, $i_G < 4^{i_L}$) is parsed and the grid mean, μ_i, is calculated. This procedure is repeated across all dies, reticles, and wafers. Finally, the standard deviation of grid mean, $\sigma_{\mu,i}$, is calculated for each grid and then entered into the corresponding row of the sensitivity matrix (as in equation 2–9).

2.3 Experimental Data and Analysis

As stated earlier, our analysis is based on 0.13μm ELM data taken from horizontal polysilicon lines (which were manufactured with typical resolution enhancement techniques such as optical proximity correction) [44]. We investigated 5 different wafers that each contained 23 fields, and each field included 308 measurement points – 14 points

in the horizontal direction and 22 points in the vertical direction. Individual measurement points were spaced horizontally by 2.19mm and vertically by 1.14mm.

An example of one wafer of ELM CD measurements is illustrated in Figure 2.3. As shown, not only do the measurements vary across the wafer (the lower right corner has smaller CD values than the upper right corner), but specific patterns occur within the reticles (the upper and lower boundaries of the field have a higher CD than the center points). For these 5 wafers, we divided the reticles into various die sizes in order to investigate the effect that die size had on CD variation. Initially, we diced a reticle into 4 die, (a 2-die x 2-die configuration where each die was approximately 15mm x 13mm). Then, we examined a number of characteristics including the mean, standard deviation, and correlation of all the dies.

The mean values for each data point in a die from the 2x2 reticle configuration are shown in Figure 2.4 (a). From this type of figure, certain trends became clear. For example, in Figure 2.4 (a) the typical die had lower values in the center of the die, and the

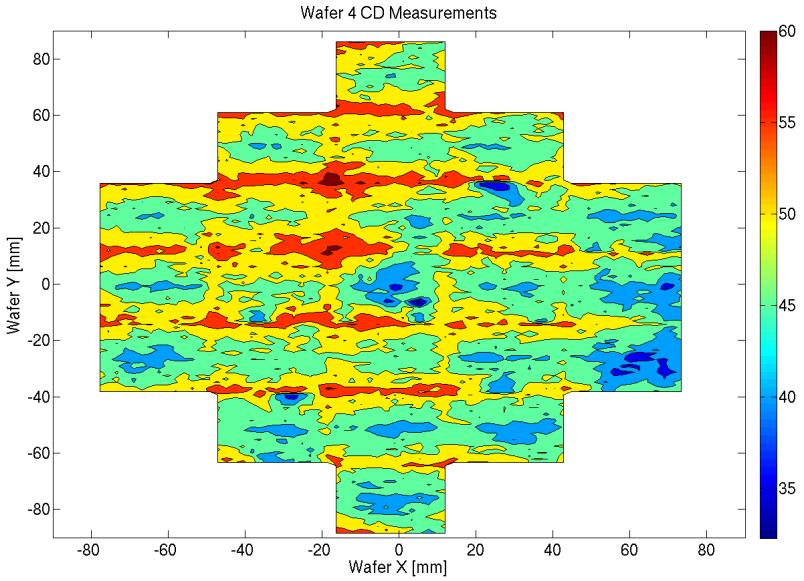
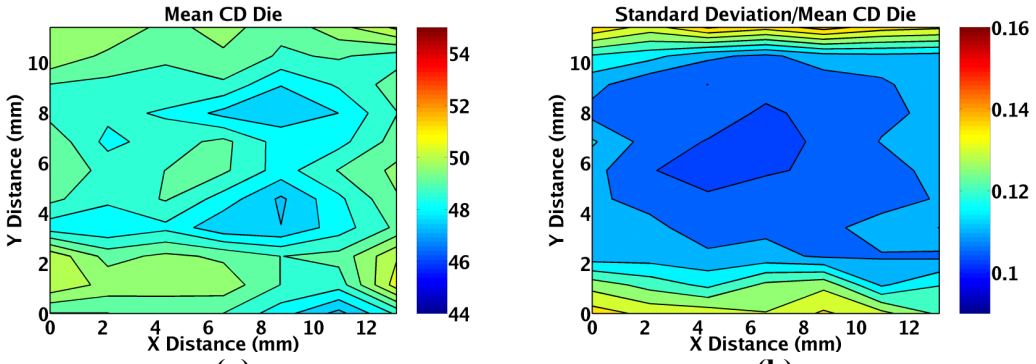


Figure 2.3. Wafer CD Measurement Contour Plot.

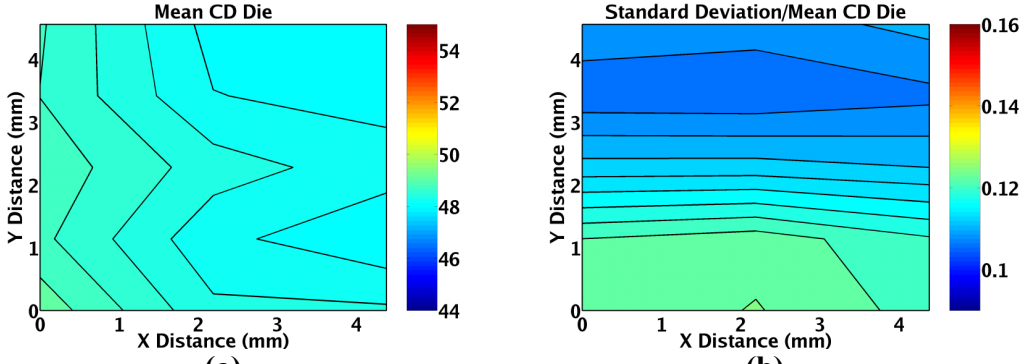
CD values increased toward the edges of the die. In Figure 2.4 (b), the standard deviation over mean is plotted for the same reticle configuration. Again, the figure shows the edge effects in the die. To contrast the 2x2 diced reticle, we have also included the equivalent plots for the 4x4 reticle configuration in Figure 2.5.

On average, the 4x4 dicing merely divided the 2x2 case into two-by-two grids of its own. Thus, it can be seen that the 4x4 mean plot is a quarter of the 2x2 plot, with the spot effect seen in the 2x2 case lying on the inner portion of the 4x4 die. Similarly, the standard-deviation-over-mean plot also resembles a quarter of the 2x2 case, with the lower deviation occurring at the top edge of the typical 4x4 die. It should be noted,



(a) **(b)**

Figure 2.4. (a) Mean CD Values for Die (2x2 reticle dice) (b) Standard Deviation/Mean for Die (2x2 reticle dice).



(a) **(b)**

Figure 2.5. (a) Mean CD Values for Die (4x4 reticle dice) (b) Standard Deviation/Mean for Die (4x4 reticle dice).

however, that the variation structure is quite different between the 2x2 and 4x4 diced cases.

In addition to the mean and standard deviation, the correlation was also extracted for different size die. Plotted in Figure 2.6 is the average correlation versus separation distance. It is easily identifiable that this function was not monotonically decreasing with distance, *x*. On the contrary, we saw many distinctive peaks where correlation fell and then rose again, sharply, at a particular distance. From this investigation, it became clear that correlation versus horizontal distance was different than the correlation versus vertical distance (i.e., correlation was typically stronger along a particular axis). This is confirmed in Figure 2.7 (a) where correlation versus distance is plotted separately for the horizontal and vertical directions.

As shown in Figure 2.7, correlation in the x-direction was actually stronger than correlation in the y-direction. We hypothesized that the reason behind this phenomenon was that during fabrication, the lithographic stepper scanned across the reticle and only printed a narrow slit in the x-direction while the entire y-dimension was printed. Thus, vertically, the reticle saw all of the variation in the lithographic system (particularly lens aberrations) but as the stepper scanned across x, the variation did not change significantly (e.g., the same part of the lens exposed all x-locations in a field), creating higher correlation in x. Figure 2.7 (b) also shows similar behavior for the smaller die size.

Lastly, we plotted probability density functions for each point within a die. One example is shown in Figure 2.8, which is a plot of point 76 within the 15mm x 13mm die (2x2 reticle) and point 14 within the 8mm x 6mm die (4x4 reticle).

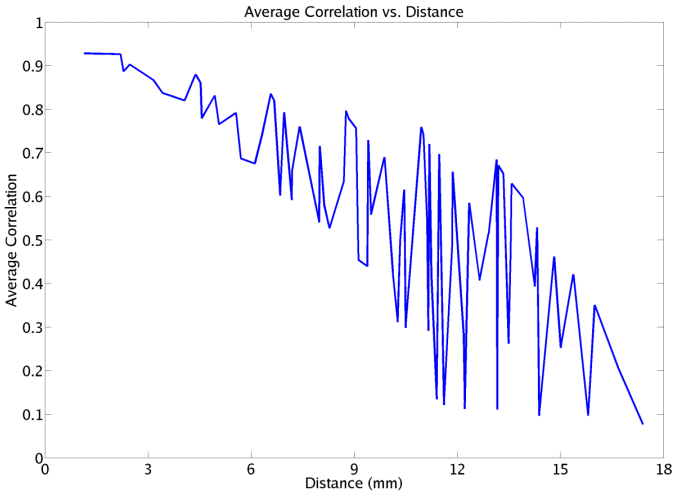


Figure 2.6. Average Correlation vs. Distance (2x2 reticle).

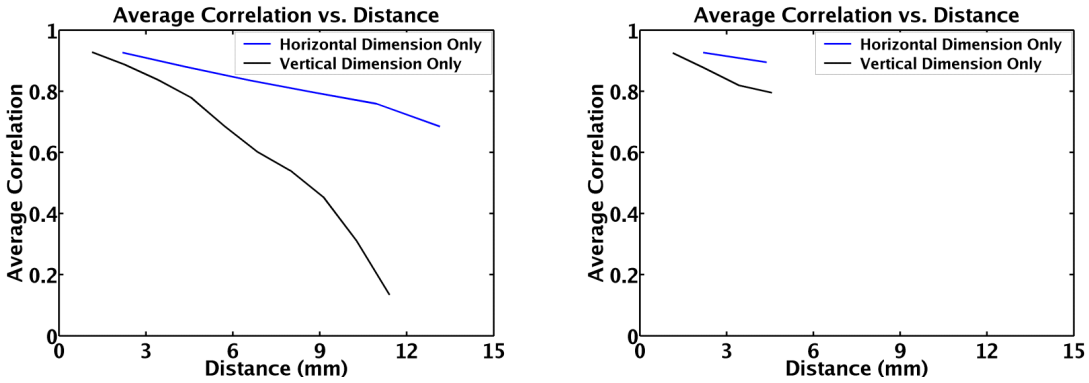


Figure 2.7. Average Correlation vs. Distance (1-dimension only).
(a) 2x2 reticle dice (b) 4x4 reticle dice

2.4 Variation Modeling and SSTA Results

After analysis of the experimental data, we used the data to test the accuracy of different correlation models and their associated SSTA runs. For our test circuit, we utilized the behavioral Verilog from an industrial 15,000 gate implementation of a Viterbi decoding circuit. Then Synopsys's *Design Compiler* was used to synthesize the design and balance the paths. Lastly, the test circuit was placed and routed using Cadence's *Silicon Ensemble*, in order to generate the placement information needed by the SSTA tool. The

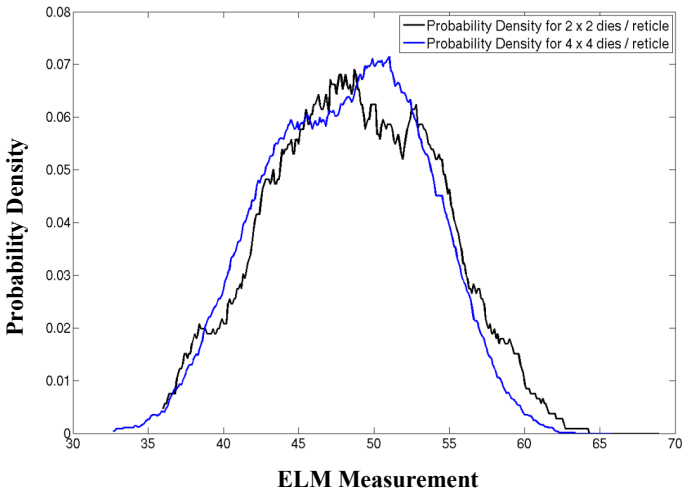


Figure 2.8. PDF Plot for ELM Measured CD.

authors would like to note that we did not actually layout the Viterbi decoder. It was merely used as a simulation benchmark to test the accuracy of our spatial correlation models.

The general flow of our analyses is illustrated in Figure 2.9. There are effectively three branches in the flow. All branches start with the same wafer data. Then, in the first case (the left branch), we perform static timing analysis on all N die, where,

$$N = X \times Y \times 23 \times 5\,, \tag{2–10}$$

X is the number of die per reticle in the horizontal direction, and Y is the number of die per reticle in the vertical direction (23 represents the number of reticles and 5 represents the number of wafers). From deterministic STA, we obtain N timing reports from which we can extract a final distribution for critical path delay of the circuit. We consider this the golden analysis of circuit delay (since it is based directly on the underlying measured data) and refer to it as the "Enumeration-based Timing Analysis" for the remainder of the chapter.

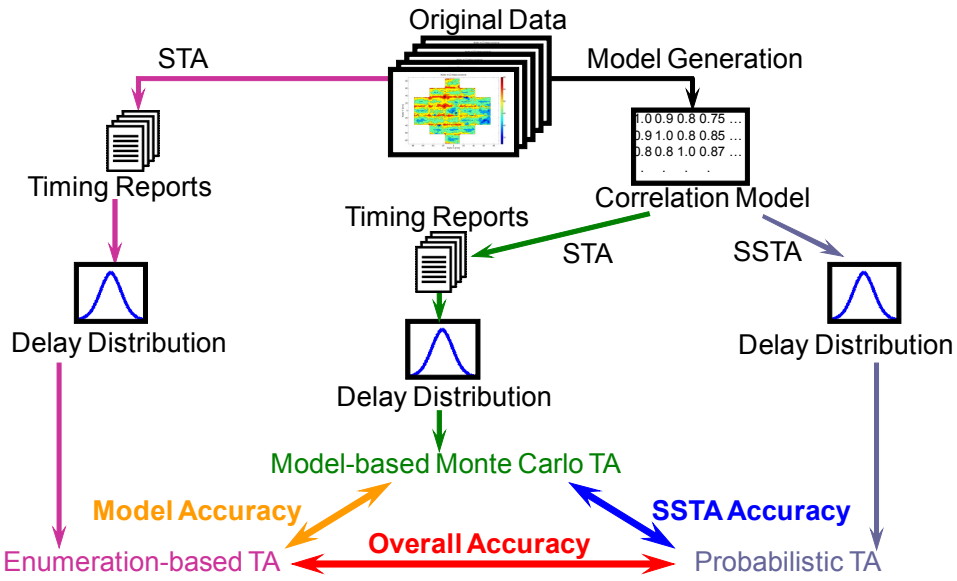


Figure 2.9. Timing Analyses Flow.

The two paths on the right both begin with a model generation step which involves fitting one of the 5 discussed CD models to the data. The two paths then diverge. The center path (referred to as "Model-based Monte Carlo") essentially follows a flow similar to the "Enumeration-based" timing analysis (TA). The only difference between the two paths is that the STA in the Model-based Monte Carlo TA is performed on random samples that were generated using the fitted correlation models, whereas the Enumeration-based TA uses the measured ELM data, directly. Finally, the right-most path, called "Probabilistic TA," performs SSTA on the fitted correlation model.

In the end, implementing this TA flow gave us three outputs available for comparison. By comparing the Enumeration-based TA with the Model-based Monte Carlo TA, we were able to determine the inherent accuracy of each correlation model. Similarly, by comparing the Probabilistic TA distribution to the Model-based Monte Carlo TA, the accuracy of SSTA, itself, was determined. Lastly, we computed the overall accuracy of

Table 2.1. Enumeration-Based, Model-Based, and Probabilistic TA Results.

Analysis Method		μ (ns)	% Error from Enum.	σ (ns)	% Error from Enum.
Enumeration-based TA		2.049	–	0.152	–
Model-based Monte Carlo TA	Die-to-Die	1.934	5.623%	0.139	8.326%
	Random	2.087	1.849%	0.058	62.12%
	D2D + Random	2.006	2.117%	0.146	3.784%
	PCA	2.033	0.800%	0.151	0.428%
	2-level Quad-tree	2.006	2.111%	0.159	4.556%
Probabilistic TA	Die-to-Die	1.945	5.108%	0.146	3.789%
	Random	2.130	3.934%	0.040	73.70%
	D2D + Random	2.006	0.769%	0.146	3.793%
	PCA	2.071	1.043%	0.148	2.694%
	2-level Quad-tree	2.061	0.577%	0.157	3.198%

using each correlation model within SSTA by directly comparing the Enumeration-based TA to the Probabilistic TA.

Table 2.1 includes the results of our TA verification flow. All three TA flow outputs are reported. Additionally, Figure 2.10 shows sample probability density plots for three of the models, including the Enumeration-based and two PCA models (Model-based and Probabilistic). All of the curves in Figure 2.10 are from the 4x4 reticle dice experiment.

When examining the Model-based Monte Carlo TA results in Table 2.1, it was clear that even the simple die-to-die models only deviated from the Enumeration-based results by less than 10%. The random model was more accurate than die-to-die with regards to the mean, but it produced considerable amounts of error in standard deviation. This was due to the fact that die-to-die variations tend to produce circuit delay variation (increasing σ) whereas random and/or spatially correlated variations tend to average out over circuit paths and, consequently, shift the mean value of circuit delay. Since the random correlation model did not model die-to-die variation, it incurred a significant error in the standard deviation of circuit delay. The "Die-to-die + random" correlation model,

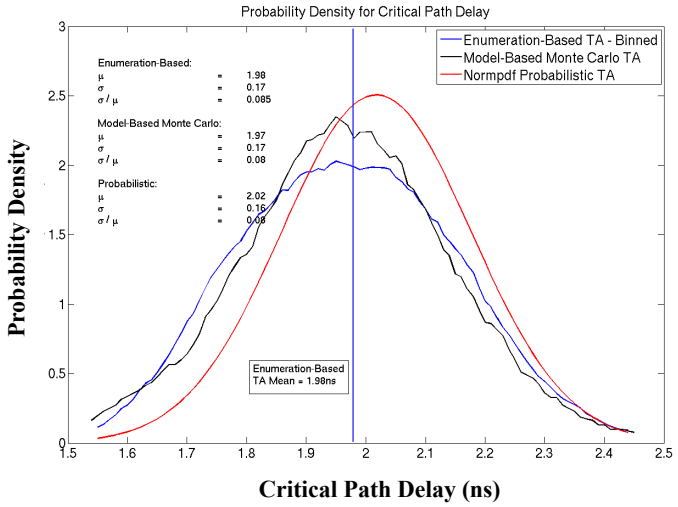


Figure 2.10. Probability Density Plots for 3 Models (Enumeration-Based, PCA Model-Based Monte Carlo, and PCA Probabilistic).

however, improved on both die-to-die and random because it modeled both components. Overall, after analyzing the simple models, it was apparent that both random and die-to-die variation were the two most important components of total variation and, of the two, die-to-die was substantially more significant.

Table 2.1 also shows the two complex spatial correlation models for Model-based Monte Carlo TA. The error in PCA was found to be negligible (falling below 1%) while the Quad-tree error was somewhat higher. The fact that the PCA correlation model out-performed the Quad-tree for Model-based Monte Carlo TA was not surprising since it utilized a much larger number of principal components to fit the measured data.

One of the more surprising results was that when we examined the error of the Probabilistic TA for the 5 models, PCA and Quad-tree reported very comparable accuracy, despite the fact that PCA reported better results for Model-based Monte Carlo TA. Both were less than 1% away from the mean of the Enumeration-based run, and both had ~3% error in standard deviation. Hence, the PCA model may have been more accurate than the

Quad-tree model, but the execution of the PCA-based SSTA incurred more error than the Quad-tree-based SSTA, making the final results approximately equal. We saw this behavior consistently across a number of different tests and postulate that this behavior was the result of the large number of independent components associated with each gate in PCA. The large number of independent components allowed PCA to obtain a better fit of the data, but also made SSTA's task more difficult and introduced a higher error in the Clark-based "MAX" function that was performed inside the SSTA tool. Finally, perhaps the most noteworthy fact gleaned from this data was that the simple "D2D + random" model performed nearly as well on the Probabilistic TA flow as the more complex models.

2.4.1 Model Accuracy vs. Die Size

Next, we studied the affect that die size had on the models and SSTA accuracies. The results are shown in Table 2.2. The cells in the first row contain the Enumeration-based results for mean and standard deviation, while the rest of the table displays the percent deviation from the Enumeration-based TA. In general the D2D, D2D + random, and Quad-tree models became more accurate (in terms of overall accuracy) as the dies decreased in size. From a D2D perspective, this was intuitive because by shrinking the die, more of the variation became inter-die variation. Furthermore, since we fit the Quad-tree to the die-to-die variation first, it followed the same trend. The random model, on the other hand, became less accurate as die size decreased because it modeled all within-die variation as uncorrelated, which was incorrect since the dies actually became more strongly correlated after shrinking, due to the inverse relationship between correlation and distance. The last model, PCA, showed a non-monotonic accuracy trend with decreasing

Table 2.2. Model vs. Die Size.

Run Type		23mmx19mm (1.2x1.2 reticle dice)		15mmx13mm (2x2 reticle dice)		8mmx6mm (4x4 reticle dice)	
		μ (ns)	σ (ns)	μ (ns)	σ (ns)	μ (ns)	σ (ns)
Enumeration-based TA		2.022	0.156	2.049	0.152	1.975	0.167
Model-based Monte Carlo TA	Die-to-Die (D2D)	4.176%	6.733%	5.281%	2.138%	2.407%	2.405%
	Random	2.136%	68.176%	1.772%	62.396%	4.545%	51.130%
	D2D + Random	0.029%	3.605%	1.105%	3.050%	0.103%	2.799%
	PCA	0.271%	6.259%	0.303%	3.472%	0.315%	1.209%
	1-level Quad-tree	3.165%	6.131%	3.098%	0.239%	0.173%	4.542%
	2-level Quad-tree	0.873%	8.979%	1.056%	1.688%	0.675%	2.039%
Probabilistic TA	Die-to-Die (D2D)	3.825%	8.492%	5.108%	3.789%	1.469%	3.192%
	Random	3.176%	83.625%	3.934%	73.703%	8.841%	62.472%
	D2D + Random	1.245%	11.247%	0.767%	3.793%	1.585%	3.188%
	PCA	0.099%	8.049%	1.043%	2.694%	2.138%	4.740%
	1-level Quad-tree	2.468%	7.451%	1.549%	1.424%	0.341%	0.280%
	2-level Quad-tree	0.794%	7.326%	0.027%	1.983%	1.002%	0.069%

die size. Using PCA on large die (i.e., die that were larger than one-quarter of the reticle) or small die (like the 4x4 reticle configuration) was less accurate than using PCA on medium-sized die. All in all, the results showed that the relative model accuracy changed based on die size and hence, different models were more appropriate in different die size scenarios.

2.4.2 Grid Model Behavior

The way in which PCA and Quad-tree behaved while varying their characteristics – such as the number of principal components for PCA and number of tree levels for the Quad-tree – was also investigated.

Limiting the number of principal components used after PCA is common practice since principal components are inherently arranged in order of decreasing importance. For our purposes, we investigated the minimum number of principal components needed to obtain accurate results from SSTA. The behavior of the mean and standard deviation of SSTA versus the number of principal components is given in Figure 2.11, and both are normalized to their respective value that includes all principal components. As you can see, both curves flatten out around 3 principal components, and approach one as the number of principal components becomes large.

Also of interest were the number of levels included in Quad-tree. However, for the tests that we ran, any number of levels above 3 did not produce noticeable gains in accuracy, since the Quad-tree SSTA already had errors of <1% for means and ~1% errors in standard deviation, as compared to the Enumeration-based model.

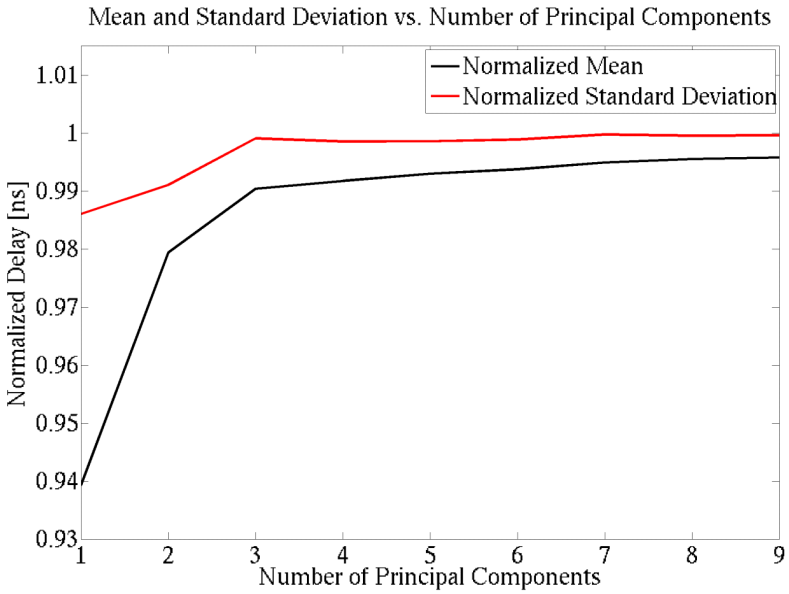


Figure 2.11. Mean and Standard Deviation vs. Number of Principal Components. (Normalized to Mean and Standard Deviation with all Principal Components)

2.5 Summary

In our analyses, we found that the grid-based models were superior, both in the Model-based simulations as well as the Probabilistic TA. On average, Quad-tree was consistently more accurate with respect to the mean, and it outperformed PCA when the die size was small. However, in all cases, the "D2D + random" model only incurred a slightly larger error (<4%) than Quad-tree and PCA. Thus, our results suggest that the "D2D + random" model can provide a simpler implementation (both in terms of overhead and run-time) while still achieving a similar accuracy range to PCA and Quad-tree, given that a certain amount of error is tolerable.

CHAPTER 3

MODELING CD VARIATION IN SSTA

To date, there has been little improvement in the delay models used within Statistical Static Timing Analysis (SSTA). This poses a potential problem, since the overall SSTA accuracy is fundamentally limited by the accuracy of the underlying models. Without sufficient accuracy, the benefits of switching from deterministic timing to SSTA are uncertain. As mentioned in both Chapters 1 and 2, of the three main variation parameters – Critical Dimension (CD), doping concentration, and oxide thickness (including t_{ox} and ILD) – CD variation modeling is particularly difficult because it contains both a systematic component that is context dependent, as well as a probabilistic component that is mainly caused by exposure and defocus variation in the lithography system. These variations in exposure and defocus create unique, transistor-specific distributions. Current SSTA frameworks, however, do not model these differences in device distributions. Instead, CD variation is handled identically across the entire standard cell library. This type of CD model is error-prone for two reasons:

- The model assumes that a single CD distribution applies to all standard cells in the library, regardless of cell type.
- The model assumes that the same, single CD distribution applies to all transistors within a standard cell.

These two assumptions lead to errors in SSTA because the resulting model does not account for the fact that different transistors (at the same location in a die) can have different CD distributions. For instance, Figure 3.1 contains a sample standard cell layout (the drawn and printed image polysilicon, as well as the diffusion layers are shown) with 12 transistors. The current CD model assumes that all 12 transistors vary identically, which means that changes in CD, or ΔCD, for each transistor can be represented by the same random variable (RV). However, in reality, each transistor CD is dependent on its neighboring geometries; the distance from neighboring gates, the distance to poly-to-contact landings (shown in Figure 3.1.B), and the line-end overhang (shown in Figure 3.1.A) will all affect an individual CD distribution. These layout characteristics not only modify the nominal CD for each device, but they also impact the variability of CD and its sensitivity to changes in lithography exposure and defocus. Thus, capturing ΔCD with a single RV is inaccurate. However, modeling each transistor CD as an independent RV is also incorrect, since exposure, defocus, and context similarities lead to correlations between CD distributions. Therefore, to accurately represent CD in a design, we would prefer a separate RV for each transistor that would not only contain the moments (μ, σ,

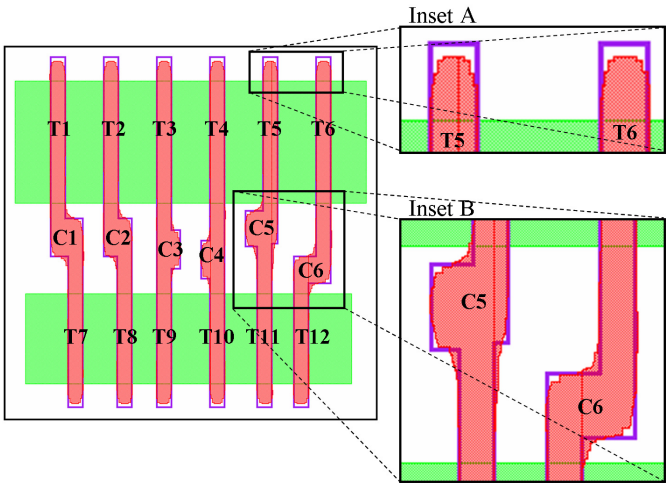


Figure 3.1. Standard Cell Layout – Poly & Diffusion Layers Only.

etc.) of its actual CD distribution, but would also preserve its correlation to other transistors.

To verify the impact of topology on both nominal CD and CD sensitivity to changes in exposure and defocus, Figure 3.2 is included, which plots CD_i (for one transistor, i, in the standard cell from Figure 3.1) as a function of lithography exposure. In Figure 3.2, four of the twelve CD_i's (T1, T2, T6, and T9) are shown. When the actual distribution of exposure is input into the CD_i function, the resulting CD distribution for transistor i has a unique mean and standard deviation, but is highly correlated to the other 11 distributions. The average CD (at each exposure setting) for the cell is also plotted and represents the single distribution CD model. Even though this is a simple example (the only transistors used to compute the average CD came from one standard cell and the only variation included was the lithography exposure variation), the single CD model still incurs an average error in standard deviation (σ) of ~9% when total variation (σ/μ) is ~4%. The zoomed in portion of Figure 3.2 emphasizes the difference in nominal CD for the transistors in the cell, as well as the difference in sensitivity (the difference in curvature) to changes in exposure.

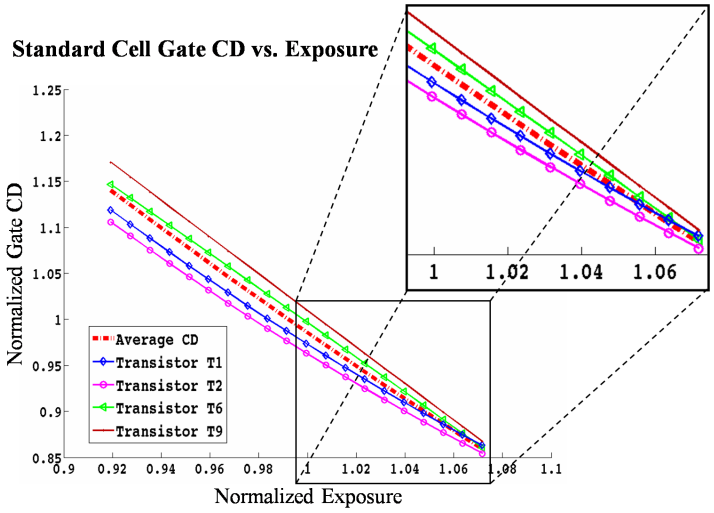


Figure 3.2. Standard Cell Gate CD vs. Exposure.

The rest of this chapter is organized in the following manner. Section 3.1 describes previous research in the field of SSTA CD modeling and has detailed descriptions on the types of models used both for CD, as well as delay. Next, Section 3.2 explains the proposed transistor-specific CD and delay models. Section 3.3 contains the experimental results obtained and Section 3.4 concludes with a brief summary.

3.1 Prior Work and Previous Approach

While there has been significant amounts of research on developing new lithography-aware characterization tools and determining how lithography impacts physical and electrical device parameters [45-47], to our knowledge, no one has proposed an accurate, transistor-specific SSTA delay model. In [45], the authors developed a lithography simulation flow which they used to improve case-based timing analysis (STA). While they showed improvement over traditional STA, it was not clear how their characterization could be extended to SSTA. An improved gate length extraction was proposed in [46] and used to improve timing accuracy in non-uniform device gates. Choi et al. in [47] designed a tool aimed at incorporating numerous sources of variation, such as proximity effects, lens aberrations, and Chemical-Mechanical Polishing (CMP). However, all the previous approaches have focused on improving STA, and are therefore applicable in the deterministic sense.

Current SSTA methodologies perform all statistical operations on propagation delays in order to determine the final distribution for timing [39,48]. However, the propagation delay for a single gate is actually a function of a number of parameters that are affected by variation (e.g., gate length and threshold voltage). In this chapter, we focus on gate length

variation. It is well known that propagation delay can be modeled as a linear or quadratic function of gate length, as shown in (3–1) and (3–2), respectively. These models typically provide a simple, but accurate, representation of delay in terms of gate length. From the models in (3–1) and (3–2), only α, β (and λ), and the distribution for L_g are needed to calculate the delay variation.

$$Delay = \alpha + \beta L_g \tag{3–1}$$

$$Delay = \alpha + \beta L_g + \lambda L_g^2 \tag{3–2}$$

In this work we chose to model delay as a quadratic function of gate length, as in (3–2), since quadratic models are capable of capturing some nonlinearity. Therefore, the delay models mentioned in the remainder of the chapter are quadratic.

While (3–2) seems simplistic at first glance, its actual implementation within timing analysis (TA) is slightly more complicated, thus, a brief description of present-day delay modeling and CD modeling follows.

3.1.1 Delay Model

Equation (3–2) is a straightforward representation of the dependence of *Delay* on one input parameter, L_g. However, in reality delay is also dependent on the output loading of the gate and the slope or slew rate of the input signal. Additionally, a gate usually has more than one input-pin, and the time it takes for an input transition to propagate to the output can vary from input-pin to input-pin. Present-day timing analysis is able to manage these dependencies by utilizing data in the form of a lookup table. This lookup table is typically built during library characterization in the early stages of a standard cell library's lifetime. For every combination of output load and input slew, the characterization tool fits

the input-to-output propagation delays as a function of gate length. Thus, for some gate in the library that has P input pins and S output-load/input-slew pairs, there will be $2 \times P \times S$ values of each coefficient: α, β, and λ (the factor of two appears because there is a rising and falling transition associated with each pin). Example pseudo-code for delay model characterization is included below and its flow diagram is illustrated in Figure 3.3.

Algorithm 3–1 DELAY_CHAR // Calculates delays for all gates

```
1:  foreach (G)  // G = gate in library
2:    foreach (p_i)  // p_i = input pin for gate, G
3:      foreach (C_L)  // C_L = output load
4:        foreach (t_slew)  // t_slew = input slew
5:          Perform transient sweep of L_g and measure delay
6:          // L_g = gate length for all transistors in gate, G
7:          Fit delay as a function of L_g
8:        end foreach (t_slew)
9:      end foreach (C_L)
10:   end foreach (p_i)
11: end foreach (G)
```

3.1.2 CD Model

The other component needed to include CD variation within SSTA is a CD model, or a model for L_g in (3–2). As stated in the introduction of this chapter, for any gate in the

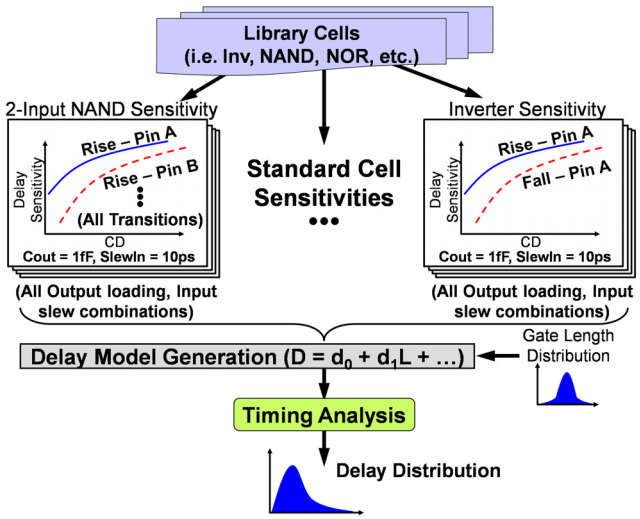


Figure 3.3. Delay Model Characterization.

library at the same location, current SSTA frameworks typically model CD as a single RV and all devices within a standard cell vary identically. Process engineers determine this distribution by fabricating different test structure geometries, and measuring the samples across a number of dies and wafers. These measurements are then treated as the discrete samples that comprise the single distribution of gate length – L_g. Once L_g is known, this model can also be extended to include spatial correlation in CD. Our SSTA implementation of this model is referred to as the "Single-CD Library" model and is discussed in more detail in Section 3.3.1.1.

3.2 Proposed Transistor-Specific Model

The probabilistic and systematic components of lithography variation due to exposure and defocus exist because of the role they play in the manufacturing process. Exposure and defocus in a lithographic system determine the amount of photoresist that is developed. Therefore, any deviation in exposure or defocus will lead to over- or under-development of the photoresist. This causes geometries to differ in stability and roughness, as well as deviate from the intended size [11,49-50]. The over- or under-development at a certain area of the die will cause probabilistic shifts in mean CD, however, the direction and magnitude of those shifts is dependent on neighborhood or context, which is systematic in nature. To illustrate this problem, we took the same standard cell (with OPC) in Figure 3.1 and ran a printed-image simulation at nominal exposure and defocus. The standard cell layout, optical proximity correction (OPC) recipe, and lithography system setup were all obtained from an industrial 90nm process. All geometries began with the same drawn CD, however, even when the printed-image

Table 3.1. Percentage Deviation from Max CD. (Nominal Exposure & Defocus)

	% Deviation from Max CD (T1)		% Deviation from Max CD (T1)
T1	0%	T7	0%
T2	4%	T8	2%
T3	4%	T9	2%
T4	4.4%	T10	3%
T5	4%	T11	2%
T6	2%	T12	3.4%

simulation was run at nominal exposure and defocus settings, context dependencies arose. Table 3.1 contains the percentage deviation of each CD from the maximum CD (the CD for the transistor labeled "T1" in Figure 3.1). From this table it is clear that even at nominal settings where OPC is typically most effective, within-cell context dependencies emerge and cause deviations in CD of ~4%. These within-cell CD deviations are caused by a number of layout characteristics (mentioned at the beginning of this chapter) like geometry-to-geometry distance, line-end overhang, and distance to contact landings. Since there are hundreds of standard cells in a typical library and each cell has different orientations/spacings of geometries, the need for a lithography-aware CD model is apparent.

Present-day, non-lithography-aware CD models can be viewed as the most rudimentary variation model: only one random variable is needed. The most complex model, on the other hand, would involve having an RV for each transistor in the library. In the 90nm library that we used, this meant that SSTA would have had to keep track of thousands of random variables for CD variation alone, which was unacceptable. However, in our work we hypothesized that since there were two main underlying components of CD variation (exposure and defocus), CD could be modeled as a function of ~2 components. Furthermore, when we performed printed-image simulations (over the entire

range of exposure and defocus) on all of the standard cells in our library, we discovered that most of the transistor CD distributions were highly correlated (>0.9), as expected, since the distributions were created by two common variation sources. These experiments suggested that a compression technique, such as Principal Component Analysis (PCA) [51], would allow us to reduce the number of RV's by >3 orders of magnitude, while still preserving the actual correlations that arose due to the common variation sources and layout similarities.

To test our theory, we used lithography-aware simulations (discussed in Section 3.2.3) to generate CD distributions for every device in our library (all transistors within every standard cell). These distributions were then treated as distinct RV's and decomposed using PCA. We determined that ~99.9% of the total variance of each RV could be captured with the first two principal components. This fact is further illustrated in Figure 3.4, which shows a scatter plot of the first 60 PCA coefficients (out of a total of ~200) for an arbitrary transistor in our library. As can be seen, the first two components are orders of magnitude

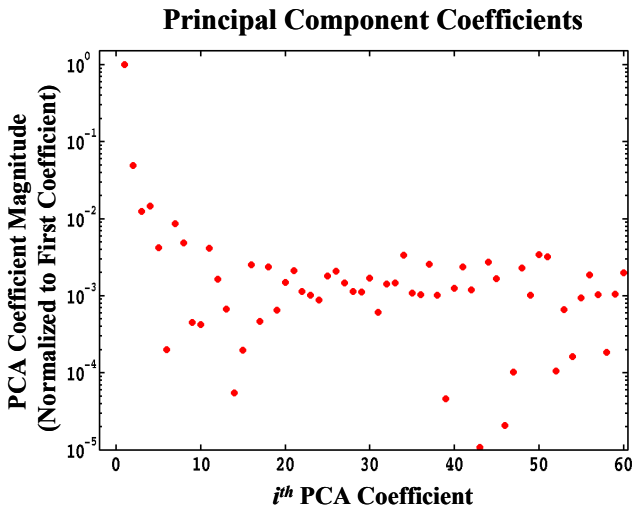


Figure 3.4. Normalized CD Distribution PCA Coefficients.

larger than the remaining components. This means that out of ~200 original RV's, only 2 are needed to accurately model CD variation for every device in our library.

The PCA compression technique is used as the basis of our Transistor-Specific (Xtor-Spfc) CD and delay models. They are described next in Section 3.2.1. Section 3.2.2 outlines the entire Xtor-Spfc characterization flow, while Section 3.2.3 briefly discusses the custom lithography-aware simulator used in our experiments.

3.2.1 Transistor-Specific CD and Delay Models

Since we use PCA to compress CD variability, the proposed Transistor-Specific CD can be analytically expressed as:

$$\begin{aligned} L_{jk} &= \mu_{L_{jk}} + a_{jk}X_1 + b_{jk}X_2 \\ a_{jk} &= \sigma_{L_{jk}} v_{jk,1}\sqrt{\lambda_1} \\ b_{jk} &= \sigma_{L_{jk}} v_{jk,2}\sqrt{\lambda_2} \end{aligned} \tag{3–3}$$

In (3–3), L_{jk} is the CD distribution of a particular transistor, j, contained in the k^{th} standard cell of the library. Specifically, $\mu_{L_{jk}}$ is the mean CD of the device (determined during Litho-Aware simulation), a_{jk} and b_{jk} are the first two PCA coefficients (calculated as described in (3–3)), and X_1 and X_2 are the principal components, which are standard, normal RV's. With respect to the a_{jk} and b_{jk} calculations, $\sigma_{L_{jk}}$ is the standard deviation of the device's CD, $v_{jk,1}$ and $v_{jk,2}$ are the jk^{th} element in the first and second eigenvectors, respectively, while λ_1 and λ_2 are the first and second eigenvalues. For a more detailed theoretical description of PCA we refer the reader to [51]. This model is referred to as the "Xtor-Spfc CD" model for the remainder of the chapter.

The Xtor-Spfc CD model is used directly in (3–2) to generate our Xtor-Spfc delay model. To determine which L_{jk} is actually used in the delay model, we merely choose the transistor associated with the specific pin-to-pin transition in question. For instance, when we characterized the rising delay transition of a minimum-sized inverter, we used the L_{jk} from the PMOS CD distribution in the delay model (and assumed single input switching). If the device happens to have multiple fingers, then we choose any one of the devices (since all of the device's CD's are highly correlated).

3.2.2 Transistor-Specific Characterization

The proposed Transistor-Specific model characterization flow is presented in Figure 3.5. It uses the Litho-Aware simulator, depicted in Figure 3.6 and described in Section 3.2.3, to determine the CD distributions for all of the transistors contained in every standard cell in our library. Then it runs PCA on the entire set of CD distributions (each CD distribution represents a distinct RV) and calculates (3–3), our Xtor-Spfc CD equation, based on the first two principal components.

We utilize the CD equations created in the flow from Figure 3.5 in two ways: we use them directly within SSTA to determine the delay distributions, and we use them to generate the gate length samples used in the Hspice delay sensitivity characterization (the L_{jk}'s are used as the L_g's in the pseudo-code in Algorithm 3–1). Because the CD distributions, the L_{jk}'s, are independent of the output loading and input slew, we only need to run the Xtor-Spfc model generation once per standard cell. When all of the CD distributions have been simulated for every cell in the library, a limited set of samples is

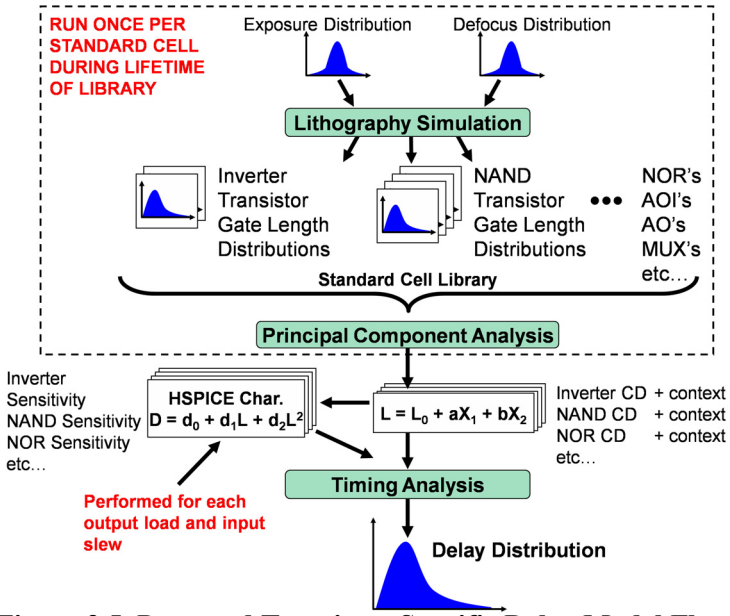


Figure 3.5. Proposed Transistor-Specific Delay Model Flow.

chosen to obtain an accurate quadratic fit for delay. As a result, the runtime of the proposed Xtor-Spfc model is on the same order as existing approaches.

It is important to note that in practice, exposure and defocus in a lithographic system gradually varies from one die location to the next. As a result, both exposure and defocus variations tend to affect closely spaced devices in a similar manner, making them more likely to have comparable CD's than those placed far apart. Therefore, it is important to capture spatial dependencies between the CD variation of two devices in addition to characterizing the proximity dependence of layout. Process engineers currently utilize test structures to determine the correlations that exist in a given process. Similarly, our model could use a test-structure-based method of extracting correlation. The test structures themselves would consist of a few representative standard cells chosen from our design library. These library cells would be replicated across the die and then fabricated at a manufacturing facility. Much like existing procedures, our RV's X_1 and X_2 would be

extracted from the manufactured data at each location in a die, across all dies, allowing both the intra- and inter-die correlation to be calculated.

3.2.3 Litho-Aware Simulation

Our Transistor-Specific characterization is built around a number of industry IC design tools. A flow chart for the simulator is shown in Figure 3.6. The Litho-Aware simulator receives a graphic data system (GDS) layout file as the main input, which contains the drawn layout of the intended design. In our library characterization, all standard cell polysilicon already had industrial OPC's, but the tool is also capable of adding corrections prior to running the printed image simulation. Next, it conditionally places neighboring geometries adjacent to all edges of the circuit under simulation so that context dependencies can be analyzed. Then, using Mentor Graphics' *Calibre*, a printed image simulation is performed on either the original GDS or the modified, context-inclusive GDS. The simulated printed image is then written to a new GDS file, which is input to an extraction tool. Finally, *Calibre* is used again, along with an industrial extraction tool, to extract the Hspice netlist and obtain actual gate length values. After

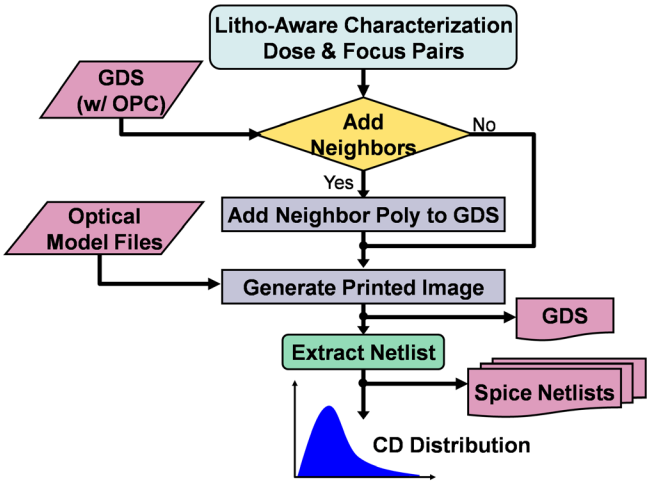


Figure 3.6. Lithography-Aware Simulator.

running this flow, there are two outputs at the user's disposal: the printed image GDS and the extracted netlists.

3.3 Results

During our library characterization, we first analyzed the gate length and delay distributions, and then explored the accuracy of three delay models: the Single-CD Library (SCDL), Cell-Specific (Cell-Spfc), and Transistor-Specific (Xtor-Spfc) models. Both the SCDL and Xtor-Spfc models were discussed previously in Sections 3.1.2 and 3.2.1, respectively. The Cell-Spfc model is a variant of the SCDL model and is described in Section 3.3.1.2. The accuracy of each of the models is found by comparing its standard deviation for delay to our "Golden" result. The Golden result for each standard cell is a discrete distribution that consists of 10,000 delay samples. Each delay sample corresponds to a printed image simulation that has been extracted and characterized in Hspice at a particular exposure/defocus setting. Each exposure/defocus pair is sampled from the joint-normal, bivariate distribution of exposure and defocus. As stated earlier, this work utilized an industrial 90nm process and an industrial lithography recipe (with industrial OPC). At the time of this work, since 90nm was a stable process and variation was expected to increase as we moved from 65nm to 45nm and beyond, we performed our library characterization, model generation, and analysis twice. In the first iteration, exposure and defocus were varied according to typical 90nm process values, but in the second iteration we increased variability so as to mimic the effects of moving from a 90nm lithographic process to 65nm. The scaling factors used to increase variability were obtained from an industry source. For the remainder of this work, we refer to the typical 90nm variation as

"90nm" or small variation and the scaled 90nm variation as "pseudo-65nm" or large variation. The authors would like to note that this experimental procedure was chosen due to the fact that the 65nm data needed for this work (standard cells, device models, and process data) was unavailable when this research was conducted.

The remainder of this section is divided as follows: Section 3.3.1 begins by describing our experimental setup. Then, Section 3.3.2 discusses the general trends observed in the CD and delay distributions, and includes a brief discussion of observed within-cell context dependencies. Lastly, Section 3.3.3 includes our model comparisons for both variability cases. Note that in either case we did not include neighborhood characterization between cells because industry sources informed us that polysilicon geometries would be more or less regular from the 45nm process node onward, reducing neighborhood effects. Thus, we left neighborhood analysis as future work.

3.3.1 Experimental Setup

Our experimental results compare three different gate delay models: the SCDL, Cell-Spfc, and Xtor-Spfc models. Refer to Section 3.2 for the details pertaining to our proposed Xtor-Spfc model.

3.3.1.1 Single-CD Library Model

For this work, we required a representative model that would demonstrate the amount of error incurred by ignoring within-cell and cell-to-cell lithography effects. This model is based on the current SSTA approach discussed in Section 3.1.2 and is referred to as the Single-CD Library model, or SCDL, for the remainder of the chapter. Essentially, our

custom Litho-Aware simulator (described in Section 3.2.3) samples a joint-normal, bivariate distribution of exposure and defocus and determines all of the transistor CD distributions for every standard cell in the library. Next, all of the samples from the transistor CD distributions are collected into one RV. This RV, L, represents the single CD distribution mentioned in Section 3.1.2, and we use the moments of L to derive L_g.

$$L_g = \mu_L + \sigma_L X_1 \tag{3–4}$$

Here, μ_L and σ_L are the mean and standard deviation, respectively, of the single gate length distribution, L, and X_1 is a standard, normal RV (with zero mean and unit variance). Finally, the delay distribution for each cell is calculated by substituting L_g into (3–2).

3.3.1.2 Cell-Specific Model

In addition to the Transistor-Specific model proposed in Section 3.2, we also explored a variant of the SCDL model, which we refer to as the "Cell-Specific" (Cell-Spfc) model. This model uses the same basic procedure described in Section 3.3.1.1, except for one key difference: instead of collecting CD distributions from the entire library into one RV, CD distributions from each cell are collected into a local gate length distribution. For example, consider the procedure that we used to characterize a minimum-sized, 2-input NAND gate that contained a total of four transistors: $NMOS_1$, $NMOS_2$, $PMOS_1$, and $PMOS_2$. After Litho-Aware simulation, all of the CD distribution samples for these four transistors were collected into one RV, L_{NAND2}, and we then calculated $L_{g,NAND2}$, as seen in (3–5), using the mean and standard deviation obtained from the L_{NAND2} distribution.

$$L_{g,\,NAND2} = \mu_{L_{NAND2}} + \sigma_{L_{NAND2}} X_1 \tag{3–5}$$

Therefore, in the Cell-Spfc model, each standard cell within the library will have a different $L_{g,CELL}$, but similar to the SCDL model, all transistors within the same cell will have identical $L_{g,CELL}$'s. These distinct $L_{g,CELL}$'s are then substituted into (3–2) on a cell-by-cell basis.

3.3.2 CD and Delay Distributions

Using our characterization tool, we analyzed 22 different standard cells under varying amounts of exposure and defocus. We discovered that with the pseudo-65nm process variation setup, our library had an average gate length distribution $3\sigma/\mu$ of ~18% and an average delay distribution $3\sigma/\mu$ of ~15%. Additionally, we verified the effect that layout topology had on the CD and delay distributions. Our experiments proved that both the CD and delay distributions were different for transistors within the same cell, as well as for transistors from two different cell types. For example, Figure 3.7 contains the probability density function (PDF) for a 4-finger, 2-input NOR gate (composed of 16 transistors total). Included in the plot are 3 of the 16 CD distributions: two NMOS and one PMOS. All three transistors are normalized to the PMOS device. From this figure, it is apparent

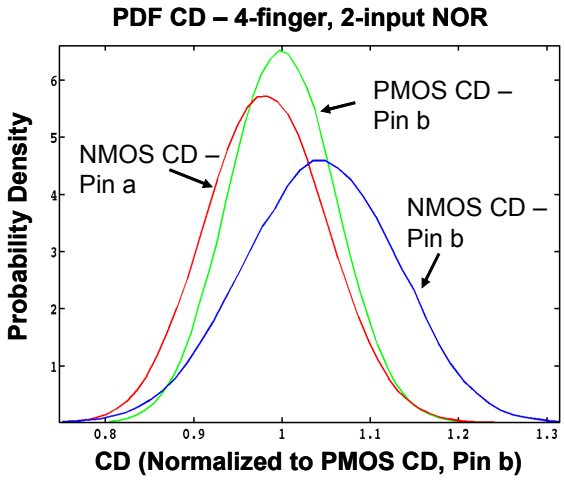


Figure 3.7. PDF for Various Transistors in a 4-finger, 2-input NOR gate.

Table 3.2. Absolute Error in Standard Deviation.
(from Golden Distribution)

	Pseudo-65nm (Avg. σ/μ = 4.9%)		90nm (Avg. σ/μ = 2.9%)	
	% Error in σ		% Error in σ	
	Rise	Fall	Rise	Fall
SCDL - Avg	10.9%	12.7%	14.3%	15.0%
Cell-Spfc - Avg	8.7%	11.4%	9.3%	9.3%
Xtor-Spfc - Avg	3.4%	4.7%	2.2%	1.4%
SCDL - WC	38.0%	39.4%	41.7%	38.3%
Cell-Spfc - WC	38.2%	39.3%	36.0%	30.8%
Xtor-Spfc -WC	16.1%	8.7%	15.4%	8.8%

that each of these distributions differ in mean and standard deviation by a few percent, thereby confirming that ignoring within-cell variation is inaccurate. The amount of inaccuracy is quantified in the following section.

3.3.3 Model Comparison

As mentioned previously, the three models discussed in Section 3.3.1 are compared in this section and each model fits delay as a quadratic function of CD, as in (3–2). We found that when comparing the three delay models to our Golden result, each model had about the same average error in mean (~1%), but the error in standard deviation (σ) differed considerably. The resulting error in σ for each model is displayed in Table 3.2. Both variation cases – Pseudo-65nm and 90nm – are included in Table 3.2, however, unless otherwise mentioned, the remaining results discussed in this chapter pertain to the Pseudo-65nm data.

From Table 3.2, it is apparent that both of our delay models, the Cell-Spfc and Xtor-Spfc, are more accurate than the current SSTA delay model, SCDL. The SCDL delay model has an average error in σ of 11.8%, and has a worst case error of 39%. Our

proposed delay model, the Xtor-Spfc model, reduces average σ error by 2.9X and has a worst case error of ~16% (a 2.4X improvement).

In order to visually portray the accuracy improvement achieved by using either the Cell-Spfc model or the Xtor-Spfc model, Figures 3.8 and 3.9 are included. These figures show the standard deviation of delay for the three models plotted against the golden standard deviation. In these plots, one point represents a model's standard deviation for one input-to-output propagation delay distribution (there are ~50 different pin-to-pin transitions for the 22 standard cells in our library). The closer a point is to the solid black line ($y = x$), where *Model* σ = *Golden* σ, the more accurate the point (and model) is. From

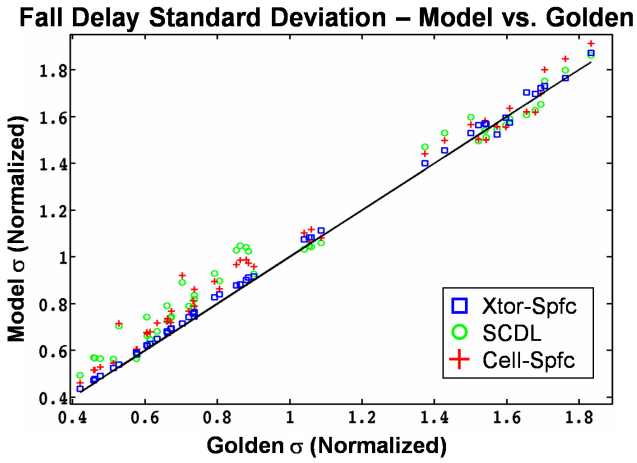


Figure 3.8. Fall Delay σ Comparison – Normalized (Pseudo-65nm Variation).

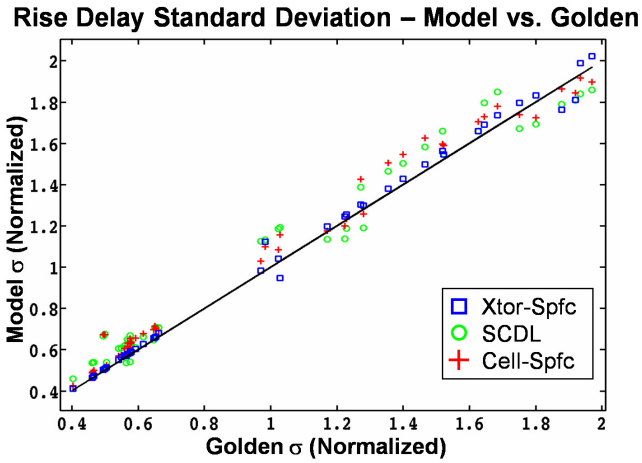


Figure 3.9. Rise Delay σ Comparison – Normalized (Pseudo-65nm Variation).

Figures 3.8 and 3.9, it is clear that the SCDL model is consistently furthest from the line, followed by the Cell-Spfc model, while the Xtor-Spfc model is the most accurate. This confirms what we observed in Table 3.2. If we look at two example CDF graphs in Figure 3.10 and Figure 3.11, we observe similar results. The Xtor-Spfc model and Cell-Spfc models follow the Golden result more closely than the SCDL model. However, here the shortcomings of the Cell-Spfc model become apparent. When we compare simple standard cell implementations, such as the minimum-sized inverter in Figure 3.10, the Cell-Spfc model is almost as accurate as the Xtor-Spfc model. But when the models are used on more complex cells, such as the AND/OR Invert gate in Figure 3.11 or standard cells with fingered transistors, then the Cell-Spfc model has nearly as much error as SCDL, since it collects many within-cell CD distributions into one RV, similar to the SCDL model.

3.4 Summary

This chapter proposed a transistor-specific CD model and its corresponding delay model. A custom Litho-Aware simulation tool was used to compare the Xtor-Spfc models to existing SSTA models and calculate the absolute error of our Xtor-Spfc CD and delay models. Our experiments suggest that the modern SSTA delay modeling approach is error-prone and can sometimes lead to twice as much error as total variation. All in all, the proposed SSTA delay model achieves average error reductions in standard deviation of ~3X when compared to current models and can be easily incorporated into existing SSTA frameworks.

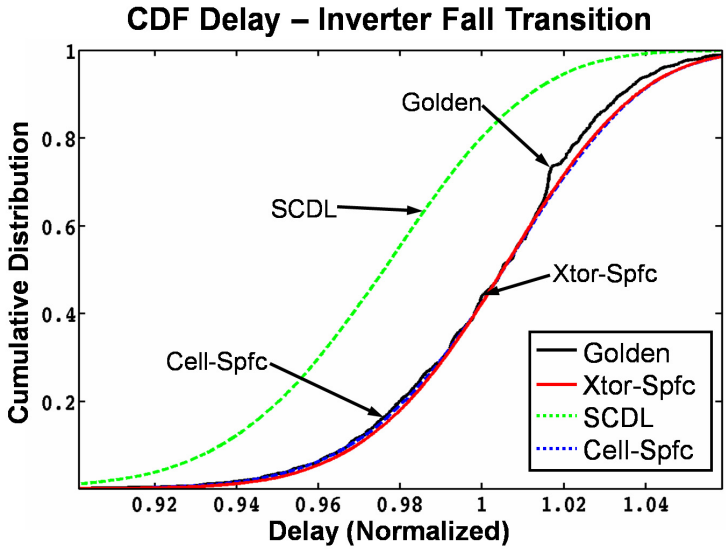


Figure 3.10. Minimum-sized Inverter Fall Delay Transition CDF (90nm Variation).

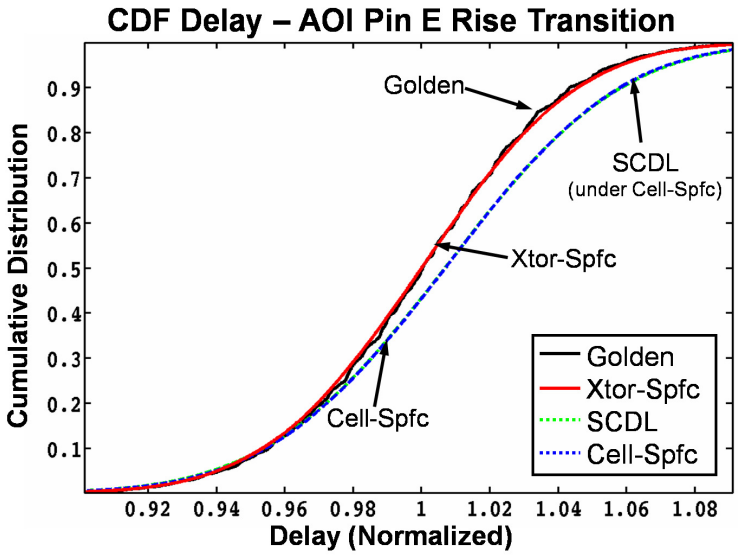


Figure 3.11. AND/OR Invert Rise Delay Transition CDF (90nm Variation).

CHAPTER 4

MECHANICAL STRESS AWARE OPTIMIZATION FOR LEAKAGE POWER REDUCTION

It was stated in Chapter 1 that as MOSFETs continue to scale below 100nm, higher effective fields cause mobility degradation, leading to decreasing drive currents. In order to battle mobility degradation and achieve higher drive currents, modern-day fabrication processes use special means to induce mechanical stress in MOSFETs, which enhances carrier mobility. Mobility enhancement has emerged as an attractive complement to device scaling because it can achieve similar device performance improvements with reduced effects on reliability and leakage.

Mechanical stress in silicon breaks crystal symmetry and removes the 2-fold and 6-fold degeneracy of the valence and conduction bands, respectively [52-53]. This leads to changes in the band scattering rates and/or the carrier effective mass, which in turn affects carrier mobility. Mechanical stress induced in a CMOS channel can be either tensile or compressive. As illustrated previously in Figure 1.4, NMOS and PMOS devices have different desired stress types (compressive or tensile) in the longitudinal, lateral, and Si-depth (vertical) dimensions. By providing the correct type of stress for a device (in one or more dimensions), we can achieve higher drain currents. However, since carrier mobility affects the drain current in all MOSFET operation regimes, increasing carrier mobility not

only increases saturation current, but it also increases subthreshold current. Specifically, short-channel MOSFET saturation drain current, $I_{D,sat}$, has a sub-linear dependence on mobility, μ_0, while the subthreshold drain current ($I_{D,sub}$) dependence on mobility is linear [17-18]. These two relationships between drain current and mobility make mobility enhancement an interesting alternative to other power/delay optimization techniques.

One of the most popular power/delay optimization techniques that has been researched considerably in both academia and industry is the dual-V_{th} optimization scheme [20-22]. This technique typically uses gate sizing and two choices of threshold voltage to optimize a given circuit for some metric (usually delay or power). Since $I_{D,sat}$ and $I_{D,sub}$ are super-linearly and exponentially dependent on V_{th}, respectively, V_{th} can potentially be a powerful optimization parameter. However, since incorporating different threshold voltages adds significant design and process complexity, practical implementations typically restrict the number of threshold voltages to ~2 [54].

One of the main disadvantages of using a dual-V_{th} scheme is, coincidentally, also one of its strengths. Since each gate in the design can either be high-performance or low-leakage, dual-V_{th} provides for a wide range of performances (due to the super-linear and exponential dependencies of $I_{D,sat}$ and $I_{D,sub}$ on V_{th}, respectively), but the approach has only coarse granularity in its selection. Mobility enhancement induced by mechanical stress, however, is layout dependent and can therefore provide much finer delay-versus-leakage control without adding to process complexity/cost. This granularity, coupled with the fact that leakage is only linearly dependent on mobility, makes stress-induced mobility enhancement an interesting research topic that can either be directly compared to dual-V_{th}

assignment, or used concurrently to provide additional gains in either leakage or delay. Since the leakage penalty incurred by mobility enhancement is significantly less than V_{th} assignment, this chapter focuses on leakage reduction. However, for completeness, the end of the chapter also demonstrates that the proposed joint optimization framework can be used to reduce circuit delay (while maintaining iso-leakage).

The remainder of this chapter is divided as follows: the first two sections, 4.1 and 4.2, describe prior mechanical stress work, our main contributions, and how they differ from previous publications. Section 4.3 includes background information relevant to mechanical-stress-based mobility enhancement and compares the power vs. performance tradeoff inherent in both mobility enhancement, as well as dual-V_{th} assignment. Next, Section 4.4 discusses the layout dependence of stress. Section 4.5 builds on the knowledge developed in Section 4.4 by presenting the stress-dependent layout properties for our 65nm technology. Results obtained by modifying these properties in 65nm industrial standard cells are discussed in Section 4.6. Section 4.7 introduces the optimization methodology used in this chapter. Lastly, Section 4.8 presents the overall optimization results and Section 4.9 concludes with a brief summary.

4.1 Prior Work

To date, most of the published work on mechanical stress in silicon has focused on the effects of Shallow Trench Isolation (STI) [33,55-56] or limited their analysis to only include the PMOS sources of mechanical stress [34,57-59]. Reference [60], on the other hand, studied variability in CMOS circuits for a low power 45nm test chip that featured STI and a tensile nitride liner as sources of stress (NMOS only). One key result

ascertained from [60] is that NMOS devices showed 5% higher performance as source/drain diffusion lengths were increased by 75%, which is qualitatively similar to the results we observed in our 65nm process that included stress sources for both PMOS and NMOS devices. In the last few years, researchers have begun exploring layout optimization techniques involving stress. For example, in [56] the authors presented an active-layer fill insertion technique which optimized circuit delay by exploiting STI stress. However, in the 65nm industrial technology used in this research, we discovered that the STI stress contribution was <10% of the total channel stress, making STI optimization less effective. The first optimization scheme developed to exploit the source/drain length dependency of mechanical stress was published in [36], which described a timing closure technique that utilized stress enhanced versions of standard cells to improve path delays. While the authors in [36] do report average delay savings of ~5%, they do not disclose the additional leakage power consumed, nor do they discuss possible leakage versus delay tradeoffs.

4.2 Contributions

The work described in this chapter differs from previously published research in that it incorporates all of the layout dependent sources of stress and, consequently, exploits a larger number of layout properties that affect stress (e.g., source/drain lengths, contact placement, distance from STI, etc.). Additionally, our optimization algorithm is not a one-sided approach that only optimizes delay. Instead, it accounts for the tradeoff between leakage and delay and achieves the largest improvement in leakage power (delay) for identical delay (leakage power). Thus, the main contribution of this chapter is a new,

circuit-level, block-based, joint optimization framework that uses stress-enhanced standard cells (in conjunction with un-enhanced cells and/or dual-V_{th} cells) to improve either leakage power consumption for iso-delay-performance or circuit delay for iso-leakage-power-consumption.

We begin by addressing the layout dependency of stress-based performance enhancement. We perform a comprehensive study in order to determine how various layout parameters affect device stress, and then analyze their impact on device performance. From this study we then extract the main layout properties that impact mechanical stress in our industrial, 65nm process. Next, these layout properties allow us to create "high-Stress" and "low-Stress" versions of a subset of standard cells from an industrial 65nm CMOS library (analogous to "low-V_{th}" and "high-V_{th}" cells in a dual-V_{th} library). Finally, we propose a stress-aware optimization algorithm and generate two comparisons: 1) stress-based performance enhancement versus dual-V_{th} assignment, and 2) combined stress-based enhancement with dual-V_{th} versus only dual-V_{th}.

4.3 Background

This section discusses the two main topics that are the foundation of this chapter: the sources of mechanical stress (and their dependency on layout properties) and how mobility and V_{th} affect drain current.

4.3.1 Mechanical Stress Sources and their Layout Dependence

Mechanical stress in silicon can be generated by either thermal mismatch or lattice mismatch. Thermal mismatch stress is caused by differences in the thermal expansion coefficient, while lattice mismatch stress is caused by differences in lattice constants. Figure 4.1 shows the major sources of stress for one of the latest 65nm CMOS technologies [61]. The sources are Shallow Trench Isolation (STI), embedded SiGe (only in PMOS devices), tensile/compressive nitride liners (in NMOS/PMOS devices, respectively), and the Stress Memorization Technique (SMT).

Shallow Trench Isolation (STI): STI creates compressive stress longitudinally and laterally due to thermal mismatch [34,56-57] and volume expansion [57]. From Figure 1.4, it is apparent that this compressive stress degrades the electron mobility in NMOS devices (in both the longitudinal and lateral directions) [62] and degrades hole mobility in PMOS devices in the lateral direction. However, STI stress that is induced longitudinally (e.g., at the left and right boundaries of standard cells) actually improves hole mobility in PMOS devices.

Embedded SiGe (eSiGe): For PMOS transistors, an eSiGe process is implemented where SiGe is epitaxially grown in cavities that have been etched into the source/drain (S/D)

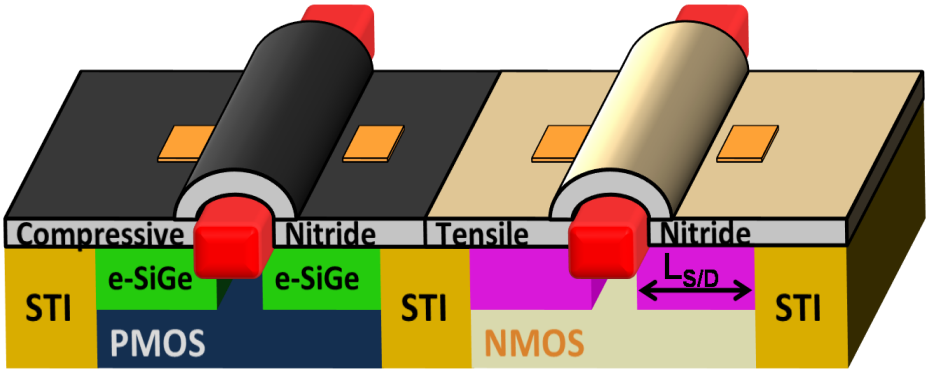


Figure 4.1. Sources of Stress for NMOS and PMOS Devices.

areas [63]. Lattice mismatch between Si and SiGe creates a large compressive stress in the PMOS channel, thereby resulting in significant hole mobility improvement.

Dual-stress Nitride Liners: As shown in Figure 4.1, mechanical stress can also be transferred to the channel through the active area and polysilicon gate by depositing a permanent stressed liner over the device [30]. Tensile liners improve electron mobility in NMOS devices, while compressive liners improve hole mobility in PMOS devices. The latest high performance process nodes have simultaneously incorporated both tensile and compressive stressed liners into a single, high performance CMOS flow, called the Dual-Stress Liner approach. In this process, a highly tensile Si_3N_4 liner is uniformly deposited over the entire wafer. The film is then patterned and etched from the PMOS regions. Next, a highly compressive Si_3N_4 liner is deposited, patterned and etched from the NMOS regions.

Stress Memorization Technique (SMT): In addition to the permanent tensile liner shown in Figure 4.1, the Stress Memorization Technique (SMT) is also used to increase the stress in n-type MOSFETs [64]. In this technique, a stressed dielectric layer is deposited over all of the NMOS regions, thermally annealed, and then completely removed. The stress effect is transferred from the dielectric layer to the channel during the anneal and is "memorized" during the re-crystallization of the active area and gate polysilicon.

A closer examination of these stress sources shows that the amount of stress transferred to the channel, and, consequently, the drive current enhancement, has a strong dependence on certain layout properties. The amount of eSiGe (and, hence, the stress), for example, depends upon the length of the active area. Longer active area also means that

the STI will be pushed further away from the channel, which will lower its effect on the total channel stress. Therefore, the drive current of a transistor depends not only upon the gate length and width (L and W), but also upon the exact layout of the individual transistor and its neighboring transistors. This means that the performance of two transistors with identical gate lengths and widths can actually differ significantly, depending on their layouts.

Beginning in Section 4.4, we study the layout dependence of stress-based performance enhancement for different device configurations and identify simple layout properties in our 65nm process that allow us to maximize the performance gains due to stress. The idea is to determine the key layout parameters that a layout designer can change to affect the transistor performance. Since we are interested in optimizing the layout, uniform techniques such as SMT can be ignored while modeling the layout dependence of stress because SMT involves a uniform film deposition, anneal and removal over all of the NMOS regions, which leads to a uniform shift in NMOS drive current that is relatively independent of layout.

4.3.2 Drain Current Dependence on Stress and V_{th}

Modifying carrier mobility directly affects the amount of current that flows between the source and drain terminals of a transistor. Increased carrier mobility increases the drain current, I_D, in all regimes of MOSFET operation, which improves transistor performance (in terms of delay) but increases leakage power. In order to study the delay-versus-leakage tradeoffs involved in stress enhancement, we examine the saturation and subthreshold current equations in order to determine their dependency on carrier mobility. This also

allows us to compare mobility enhancement to other performance enhancement techniques, such as V_{th} reduction. Equations (4–1) and (4–2) below give the expressions for drain current when the transistor is operating in the saturation and subthreshold regimes, respectively [17-18].

$$I_{D,sat} = \frac{\mu_0}{[1 + U_0(V_{GS} - V_T)]} \cdot \frac{C_{ox}}{2aV} \cdot \frac{W}{L_{eff}} \cdot (V_{GS} - V_T)^2$$
$$V = \frac{1 + v_c + \sqrt{1 + 2v_c}}{2} \qquad v_c = U_1((V_{GS} - V_T)/a) \tag{4–1}$$

$$I_{D,sub} = A \cdot e^{\frac{1}{n v_T} \cdot (V_G - V_S - V_{th0} - \gamma' V_S + \eta V_{DS})} \cdot (1 - e^{(-V_{DS})/v_T})$$
$$A = \mu_0 C_{ox} \frac{W}{L_{eff}} v_T^2 e^{1.8} e^{-\frac{\Delta V_{th}}{\eta v_T}} \tag{4–2}$$

From (4–1) and (4–2), it is evident that the saturation drain current ($I_{D,sat}$) has a sub-linear dependence on mobility, μ_0 (due to the vertical field mobility degradation coefficient, U_0) while the subthreshold drain current ($I_{D,sub}$) dependence on μ_0 is linear. The drain current dependence on V_{th}, however, is almost linear in saturation, but is exponential in the subthreshold regime. Therefore, if we obtain identical saturation current improvement using two separate enhancement techniques: 1) stress-based mobility enhancement, and 2) V_{th} reduction, then the corresponding increase in leakage current for the reduced-V_{th} case will be much higher (due to the exponential dependence of $I_{D,sub}$ on V_{th}). Consequently, the reduced increase in leakage current makes mobility enhancement a more attractive option than its V_{th} counterpart.

The benefits of using mobility enhancement over V_{th} reduction is illustrated in Figure 4.2, which shows the normalized I_{on} versus I_{off} curves for stress-based and V_{th}-based performance enhancements for an isolated, 65nm PMOS device. The device has three sources of stress: STI, a compressive nitride liner, and eSiGe source/drain regions. Stress is varied by changing the active area length, while the n-channel doping is changed to vary V_{th}. The curves clearly show that the tradeoff is better for stress variation. For a 12% improvement in I_{on}, the leakage for the V_{th} case is nearly twice as large as that for the stress-based improvement (shown in Figure 4.2 as points P1 and P2), and the difference is only amplified for higher values of improvement. Also, stress-based improvement allows for more fine-grain improvement control than V_{th} assignment, given that only 2-3 V_{th} values are typically allowed. Therefore, a designer would prefer to achieve performance improvements through stress-enhancement whenever possible, due to the reduced leakage penalty and increased granularity. The superiority of the stress-based performance improvement technique makes it an appealing option for further investigation. Thus, the

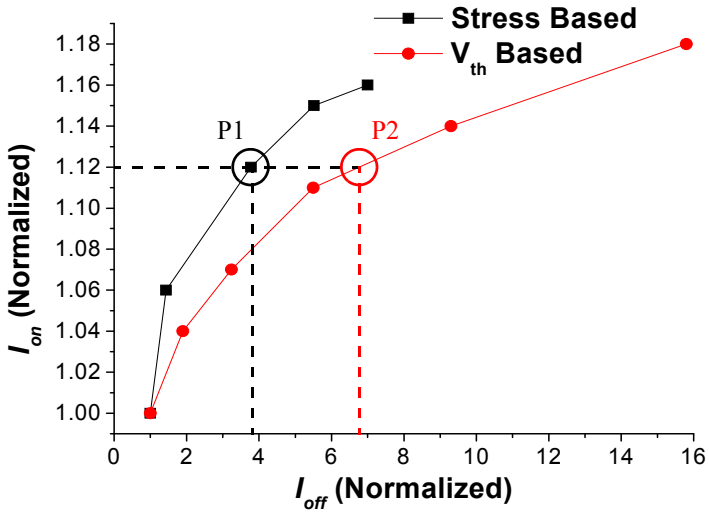


Figure 4.2. 65nm PMOS I_{on} vs. I_{off} for V_{th}-based and Stress-based Enhancement.

next two sections study the layout dependence of stress, and identify the primary layout properties that can be modified so that stress-induced enhancements are maximized.

4.4 Layout Dependence of Stress-Based Performance Enhancement

In order to study the layout dependence of stress-based performance enhancement, we used the *Davinci* 3D TCAD tool [65], which has an extensive set of stress-related features. Additionally, we followed the layout rules from an industrial 65nm CMOS technology and the device fabrication was simulated in *Tsuprem4* [66] (in order to capture the process-induced stress). The stress values were then imported into *Davinci*, which simulated the device and solved for the stress-based mobility enhancement equations. The resulting values for drive current and leakage were verified against experimental test chip data, which was consistent with previously published 65nm technology data for minimum sized NMOS and PMOS devices [61]. Furthermore, the simulated values of stress were in close agreement with previously reported data for PMOS channel stress while considering all of the layout dependent sources of stress [63]. Due to the absence of any previously published data on the layout dependence of stress or drive-current (due to stress), measured test chip results were used to quantify the impact of layout diversity on device performance. The fabrication process used for this test-chip employed all the known stress enhancement techniques. The hardware data was used to verify the accuracy of our TCAD setup, and the TCAD-based simulation results were found to be in close agreement with the measured data. Our consistency with these fabricated measurements can be attributed to the fact that we model all of the layout dependent sources of stress in the industrial

65nm technology. For a PMOS device, the sources of stress that are layout dependent include the compressive nitride liner, eSiGe, and STI. The NMOS sources, on the other hand, only include the tensile nitride liner and STI. We have ignored the Stress Memorization Technique (SMT) in our simulations, since it involves a uniform deposition and eventual removal of a dielectric layer over all NMOS devices (as discussed previously in Section 4.3.1). SMT, therefore, does not depend on layout properties and can be accurately treated as a uniform increase in NMOS drive current, independent of layout [67].

Previously, Figure 4.1 showed the 3D cross-section of an isolated PMOS device surrounded by STI. For the device shown, we increase the active area length ($L_{S/D}$) and examine the corresponding changes in drive current.[1] Increasing active area length has a number of effects: 1) it increases the amount of eSiGe, causing more stress to be transferred to the channel; 2) it increases the distance between the channel and the STI, decreasing the effect STI has on channel stress; and 3) it allows more nitride over the active area. The nitride layer actually transfers stress in two ways – vertically through the gate and longitudinally through the active area. Since active contacts create openings in the nitride layer, the longitudinal component of nitride stress can be increased by moving the contacts away from the channel. Similarly, a source/drain region that does not have any contacts (or has a smaller number of contacts) will have higher channel stress than one that has a high contact density.

[1] The authors would like to note that in this document, $L_{S/D}$ is equivalent to both the $L_{S/D}$ and $L_{p/p}$ used in previous works (such as [36]). Thus, for the remainder of the document, $L_{S/D}$ can refer to any longitudinal S/D dimension.

Figure 4.3 (a) shows the longitudinal stress (S_{xx}) in the same isolated PMOS device for two normalized $L_{S/D}$ values of 1 and 1.58 (the values are normalized to the length of a minimum-sized, contacted S/D region). Figure 4.4 shows the PMOS drive current, I_{on}, and leakage current, I_{off}, plotted against $L_{S/D}$, while Figure 4.5 shows the normalized PMOS longitudinal stress plotted against $L_{S/D}$. Results show that for a 12% performance increase, leakage current only increases by 3.78X. This I_{on} versus I_{off} tradeoff is much better than the tradeoff produced by the alternative, V_{th}-based enhancement technique, as predicted in Section 4.3.2. Additionally, Figure 4.4 shows the saturation point for extending $L_{S/D}$. Increasing the S/D length beyond 1.58 (normalized) yields minimal performance gains, even when active area length and leakage current are increased substantially. Finally, the performance enhancement is also sensitive to contact placement. Moving the contacts away from the channel accounts for nearly 2.6% of the drive current improvement and a device with a non-contacted drain (typically seen in series devices) has ~4% higher performance.

Unlike its PMOS counterpart, NMOS device performance is actually degraded by STI since STI induces compressive stress in the channel. Thus, increasing NMOS $L_{S/D}$ not

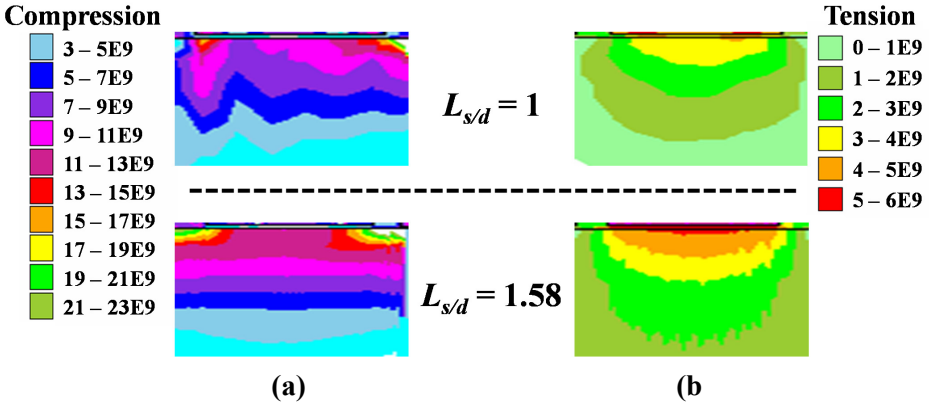


Figure 4.3. Longitudinal Stress, S_{xx} (Pa), for Normalized $L_{S/D}$ of 1 and 1.58. (a) PMOS (b) NMOS

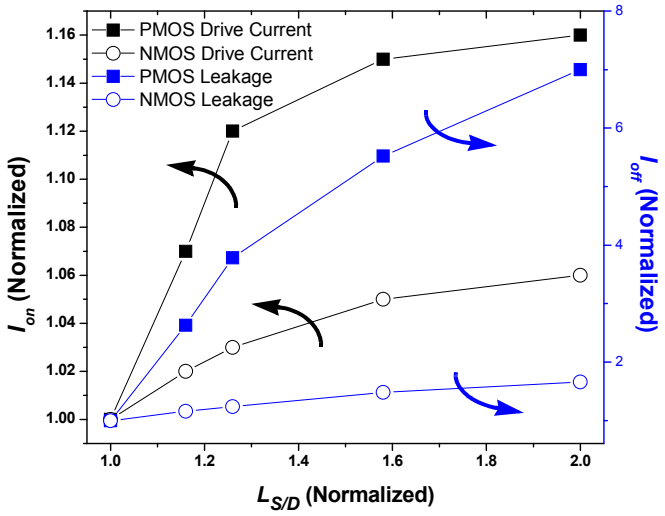


Figure 4.4. I_{off} and I_{on} vs. $L_{S/D}$ for Stress-based Enhancement in Isolated PMOS and NMOS Devices.

only pushes away the compressive STI, but it also allows for more contact separation from the channel. Figure 4.3 (b) shows the longitudinal stress in an isolated NMOS device for normalized $L_{S/D}$ values of 1 and 1.58. In addition to PMOS I_{on} and I_{off}, Figure 4.4 also shows NMOS I_{on} and I_{off} while Figure 4.5 shows its normalized longitudinal stress versus $L_{S/D}$. For NMOS devices, a 5% performance gain can be achieved for a 1.48X increase in leakage current. NMOS devices also have the same (normalized) upperbound for $L_{S/D}$

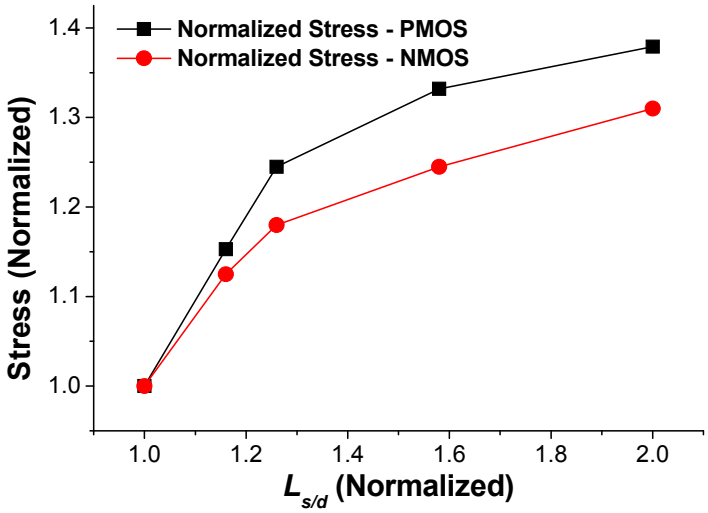


Figure 4.5. Longitudinal Stress vs. $L_{S/D}$ for Isolated PMOS and NMOS Devices.

extension as their PMOS counterparts, 1.58. Beyond this value, the area and leakage current penalties do not warrant the minimal gains in I_{on}. The increase in performance in NMOS devices, however, is limited by the fact that we are only increasing the nitride's longitudinal stress through the active area (about 35% of the total stress due to the nitride liner), and pushing away the STI (which has a relatively smaller contribution to the overall channel stress). Experimental results show that almost 80% of the total NMOS improvement is due to moving the contacts and a device with a non-contacted drain has ~2% higher performance.

Next, we studied transistor performance in denser layouts. Figure 4.6 shows the channel stress and the corresponding layout view for three PMOS transistors in a 3-input NAND gate. The device in the center (device 2) has higher stress than the two corner transistors because it is surrounded by more eSiGe (its own S/D regions as well as its neighbors' S/D regions). This difference in stress is reflected in their drive current performance, and simulations show that the drive currents for the center and edge devices differ by 8.2%. Furthermore, if there were five devices side-by-side instead of three, the difference would increase to 14.8%. This means that the drive current of a transistor is not

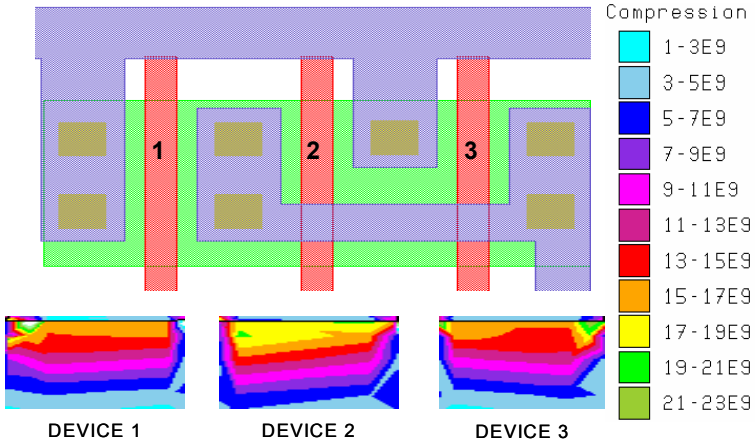


Figure 4.6. PMOS Devices in a 3-input NAND and their Channel Stress Contours (Pa).

only layout-dependent, but it is also location-dependent. Similar experiments for NMOS devices show differences of 7.4% and 12.2% for the case of three and five side-by-side transistors, respectively.

4.5 Layout Properties that Impact Mechanical Stress and Performance

Based on the intuition developed in the previous section, we now identify 3 simple layout properties in our 65nm technology that can be used to optimize a given layout for stress-induced performance enhancement. Once the properties are presented, the end of this section discusses one other important stress effect: the position-dependency of stress-induced performance enhancement. When mechanical stress is present in MOSFETs, matching W and L does not guarantee similar transistor performance even when neglecting process variation. Apart from W and L, the drive current is also affected by the layout parameters that influence stress: active area length, placement and number of contacts, and device context (i.e., whether the device is surrounded by other transistors or isolated by STI on one or both sides). In this chapter, we have already discussed the first two parameters in great detail, while the third parameter (device context) has only been briefly mentioned (at the end of Section 4.4). However, since the device context or position of a transistor within a layout also affects performance, it must be accounted for by the designer, so this phenomenon is discussed in more detail at the end of the section.

Upon finishing the layout dependency study in Section 4.4, we determined that in our 65nm industrial process, the following 3 properties had the largest impact on improving performance (without modifying existing cell boundaries).

Layout Property #1:*Active Area or Source/Drain Lengths*

Using the length of a transistor's source or drain regions (or, equivalently, changing the amount of active/diffusion area) to modify stress-enhancement is well known technique and has been studied in a number of works [34,36,59-60]. Increasing the active area moves the STI regions away from the channels and increases the amount of eSiGe in PMOS devices. Moving the STI farther from the channel improves the performance of NMOS devices since STI exerts a compressive stress in the longitudinal direction, which degrades the NMOS electron mobility. For PMOS devices, on the other hand, compressive STI stress is actually beneficial and improves hole mobility. However, increasing the active area for PMOS devices still results in higher stress due to the relatively small contribution of STI compared to the other sources of stress. Measurements show that the stress due to STI represents <10% of the total channel stress. Therefore, the increase in eSiGe and its resulting contribution to PMOS channel stress dominates the stress due to STI and provides a significant increase in hole mobility.

Increasing the active area can most readily be accomplished in a compact pull-up or pull-down network (often containing an NMOS or PMOS stack) that does not use the full width of a cell (Figure 4.7 shows the scope for increasing the active area of a PMOS stack in a 3-input NOR gate). In the case of stacked transistors, the layout does

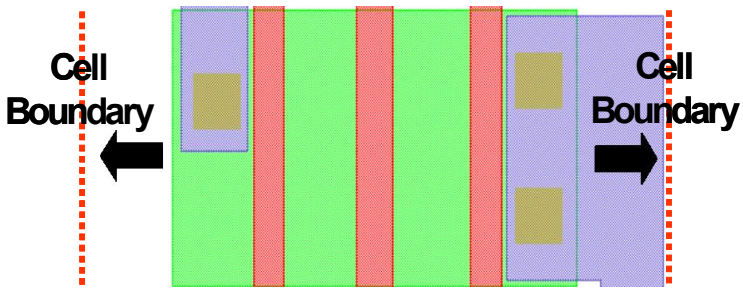


Figure 4.7. Application of Layout Property #1 to PMOS Stack in 3-input NOR.

not require contacts between intermediate nodes. Thus, their spacing can be significantly tighter because nodes that contain contacts need larger spacing to satisfy the technology's design rules. In the absence of stressors, it is best to minimize the active area in order to reduce the capacitance. However, in the presence of stressors, increasing active area length also results in higher stress in the channel (and, hence, higher drive current), in addition to increasing the source/drain capacitances. In a given CMOS layout, increased S/D capacitance for transistors closer to the output will directly affect the output capacitance, while transistors closer to the VDD and VSS rails will have a smaller effect. Hence, this layout property should be increased in cells with larger output loads, so that the change in capacitance is a small fraction of the total output capacitance. The authors would like to note that the mechanical stress dependence on active area can also be exploited to create high performance versions of standard cells which incur some area penalty, but are assigned optimally within a design.

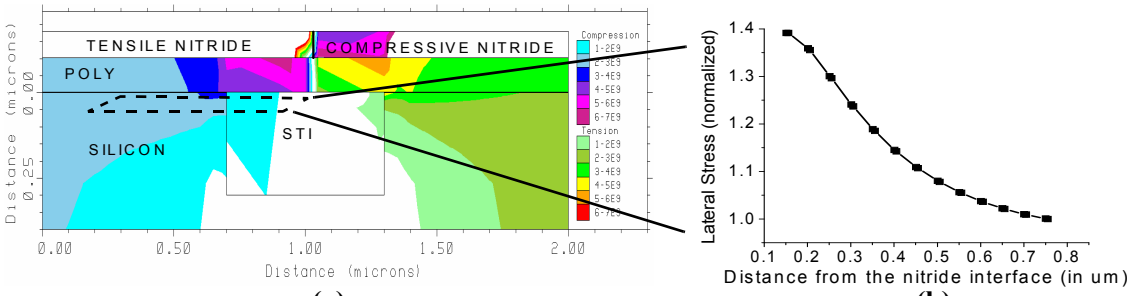
Layout Property #2:*Contact Placement*

Moving the contacts away from the channel allows more stress to be transferred by the nitride layer. For isolated devices, pulling the contacts as far away from the gate polysilicon as the design rules permit maximizes the stress-enhancement. Contacts between two gates, on the other hand, can either be placed midway for identical performance enhancement of both transistors, or placed closer to the non-critical transistor (increasing stress in the critical device). Moving the contacts away will also result in a small increase in the source/drain resistance, but, in our 65nm study, this increase was typically less than 5Ω (based on sheet resistance calculations for the maximum S/D dis-

placement obtained while creating the stress-aware optimized library), and the resulting gain in drive current outweighed the increase. The maximum S/D contact displacement observed was 60nm.

Layout Property #3:*Lateral Active Area Placement*

From Figure 1.4, we know that the desired stress in the lateral direction is tensile for both NMOS and PMOS devices. Figure 4.8 (a) shows the lateral stress behavior near the interface of the two nitride layers (cross-section across the poly going from PMOS to NMOS over STI). Figure 4.8 (b) shows the plot of normalized lateral stress (normalized to the stress value at the point farthest from the nitride liner interface) at a depth of 1nm below the Si surface versus the distance from the tensile/compressive liner interface, under the tensile nitride layer. The behavior is interesting in the sense that there is a region of compressive stress under the tensile nitride (the NMOS side) and there is a region of tensile stress under the compressive nitride (the PMOS side). This behavior follows from the physics involved behind the stress-inducing process step. At the compressive/tensile nitride liner interface, each nitride layer exerts an equal and opposite force on the other nitride layer, which imposes the opposite type of stress under the adjacent layer. Therefore, if possible, it is beneficial to move the PMOS active area into



(a) **(b)**

Figure 4.8. Stress (Pa) at Nitride Interface for NMOS and PMOS.
(a) 2D view Across Lateral STI (b) Behavior Under Tensile Nitride at Channel Depth

this region of tensile stress and the NMOS away from the region of compressive stress. The space for this movement is most readily available when the transistor widths are small but the cell pitch (lateral size) is large (due to pitch uniformity across standard cells). This combination of properties, for example, is common in minimum sized, simple gates (e.g., minimum size inverters, buffers, or 2-input NAND/NOR's).

It should be noted that the lateral active area placement will slightly alter the V_{th} of the shifted devices, due to well edge proximity effects [68-70]. However, since the amount of lateral shift applied to the 65nm standard cells was <0.205μm for the NMOS cells and <0.12μm for the PMOS cells, the corresponding shift in V_{th} was found to be <0.32mV (in both Hspice and TCAD simulations, independently) for all devices.[2] Since this V_{th} shift is relatively small, the reported results described in the remainder of the chapter do not include the well edge proximity change induced by Layout Property #3. However, if this shift in threshold voltage becomes appreciable in future processes, our experimental setup can easily be modified to include a well edge proximity model, such as the ones described in [69-70], which will capture the corresponding change in V_{th}.

Apart from these three layout properties, a designer must also be aware of how the channel stress is affected by the position of a device within the layout. Stress in the channel of a device depends not only upon its S/D lengths and contact placement, but also upon its surroundings. As we have shown in the previous section, devices that share their

[2] Hspice well-edge proximity was captured during Calibre PEX parasitic extraction, and then fed into our industrial BSIM models to calculate the effect on V_{th}. Note that the 0.32mV shift reported can be viewed as the shift in ΔV_{th} (the change in V_{th} due to well proximity), not total ΔV_{th} itself.

source/drain regions with other transistors have significantly higher stress (and hence drive current enhancement) than those at the edges of an active region (which are therefore bordered by STI), even for identical $L_{S/D}$ and contact placement. This difference in stress can be attributed to the effects of STI, as well as the fact that stressors for a device also affect its neighbors.

Ignoring the position-dependence of stress could lead to a number of design issues. First of all, the location of a transistor could result in an unexpected increase in drive current, resulting in smaller delay and possible hold-time violations, as some gates might be faster than expected. Secondly, the position-dependent current offset could modify the noise margins of a circuit. Hence, for circuits that are sensitive to noise margins (e.g., SRAM cells, Sense Amplifiers, etc.), these deviations must be accounted for either during the design phase (for example, by guardbanding against position-dependent offsets), or during the layout phase (e.g., by modifying the $L_{S/D}$'s to cancel the offsets). Finally, in certain circuits, if the strength of a transistor (in terms of drive current) is increased beyond the expected value, it could cause a substantial drop in performance. A detailed example of context-sensitive design is included in Section 4.6. All in all, designers need to be aware of the effect that position has on performance, especially if pin-to-pin delay, noise margins, or transistor strength are essential to a particular design.

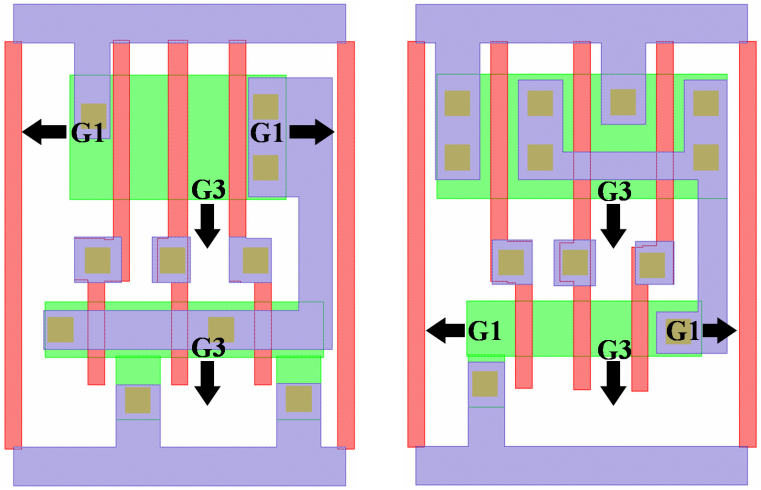
There are three main ways that a designer could capture the position dependence of stress within a particular design: fabrication, TCAD simulation, and electrical circuit simulation. The first solution, fabrication, is an expensive and time consuming endeavor, especially during the early stages of a process's lifetime. The second alternative – using TCAD tools to simulate the position dependence of stress – can be costly in terms of

runtime, and convergence becomes extremely difficult when simulating more than 10 devices at once. The final solution, electrical circuit simulation (e.g., Hspice simulation), promises to be the most efficient in terms of both cost and runtime. Unfortunately, to our knowledge, there has been little research dedicated towards electrical models that capture the layout dependence of stress. Furthermore, of the few that have been published (such as [58]), none have been implemented within an electrical circuit model (e.g., BSIM). The problems associated with each of these solutions make modeling the position dependence of stress an important and interesting research topic that remains largely unexplored.

4.6 Modifying 65nm Standard Cell Layouts

This section discusses the effectiveness of modifying the layout properties from Section 4.5 in standard cells from an industrial 65nm CMOS technology library. For a given layout, as shown in Section 4.4, a basic tradeoff always exists between the source/drain length, $L_{S/D}$, and the improvement in drive current. By exploiting this tradeoff, we can make faster, but leakier, versions of the standard cells with varying area increments and assign them intelligently to the critical paths in order to optimize performance. The performance enhanced versions all use a combination of the three properties discussed in Section 4.5: increased $L_{S/D}$, larger poly-to-contact spacing, and stress-aware lateral placement.

For example, Figure 4.9 (a) shows the layout for a 3-input NOR gate. It consists of three PMOS transistors in series (a 3-PMOS stack) and three NMOS transistors in parallel. This means that the source and drain of each NMOS is connected to the ground and the



(a) (b)

Figure 4.9. Two Layouts Illustrating Scope for Layout-based Stress Improvement. (a) 3-input NOR Gate (b) 3-input NAND Gate

output, respectively, necessitating contacts at each node. The PMOS stack on the other hand, only needs one contact to V_{DD} (at the source of the leftmost PMOS) and one contact to the output (at the drain of the rightmost PMOS). Using the classical layout methodology (where stress is ignored and capacitance is minimized), we can shrink the non-contacted S/D regions to lower the parasitic PMOS capacitance. As shown in Figure 4.9 (a) (labeled "G1"), the PMOS region has the capability of increasing the source/drain lengths (Layout Property #1) by ~22% without affecting the overall cell area. While increasing the source/drain lengths, we simultaneously shift the contacts away from the gates (Layout Property #2), maximizing performance enhancement. If we increase the active area uniformly for all transistors, drive current improves by ~12% for each PMOS device. Also, there is lateral room to move the NMOS and PMOS active area and exploit the stress dependence of Layout Property #3 (labeled "G3" in Figure 4.9 (a)). This leads to further improvements of about 3% and 1.5% for NMOS and PMOS devices, respectively. Therefore, for the 3-input NOR gate, we observe overall improvements in drive current of

Table 4.1. Percentage Contribution of Layout Properties 1–3 to the Overall Drive Current Improvement for PMOS/NMOS Stacks.

	Property 1	Property 2	Property 3
NOR3 PMOS	69.6%	19.3%	11.1%
NAND3 NMOS	20.1%	37.8%	42.1%
NOR2 PMOS	53.3%	26.6%	20.1%
NAND2 NMOS	10.1%	27.2%	62.7%

~13.5% for PMOS devices and ~3% for NMOS devices. Similarly, by modifying Layout Properties 1–3 in a 2-input NOR gate, we can achieve drive current improvements of 7.5% and 3% for the PMOS and NMOS devices, respectively.

Similarly, Figure 4.9 (b) shows the layout for a 3-input NAND gate. Instead of a PMOS stack, there is an NMOS stack in the NAND gate, so there is a potential to increase the NMOS active area length without affecting the cell area. While altering Layout Properties 1 and 2, we obtain an improvement of ~4% for each of the NMOS drive currents. Also, there is space for moving the active areas to exploit the mobility dependence of Layout Property #3. This leads to further improvements in NMOS and PMOS devices of ~3% and ~1.5%, respectively. Overall, we can achieve a ~7% NMOS performance enhancement and a ~1.5% PMOS performance enhancement. Similarly, by modifying Layout Properties 1–3 of a 2-input NAND, we can obtain drive current improvements of 4.5% and 1.5% for the NMOS and the PMOS devices, respectively. Scope for such layout-based improvements is found in most of the standard cells in our library.

Table 4.1 shows the percentage contribution of each layout property to the total drive current improvement achieved for PMOS and NMOS stacks in 2- and 3-input NOR and NAND gates, respectively. The relative contribution of the properties varies between the

Table 4.2. Summary of Stress-Aware Layout Optimization Drive Current Improvement and Tradeoffs in 65nm Standard Cells.

Cell Name	Drive Current Increase (%) after Layout Optimization		Leakage Current Increase after Layout Optimization		Leakage Current Increase after Vth Reduction (iso-drive current)		Output Capacitance Increase (%) (FO4 output loading)
	NMOS	PMOS	NMOS	PMOS	NMOS	PMOS	
3-input NOR	3%	13.5%	1.22X	4.02X	1.31X	9.20X	2.74%
2-input NOR	3%	7.5%	1.22X	2.24X	1.31X	3.52X	1.92%
3-input NAND	7%	1.5%	1.98X	1.10X	2.36X	1.53X	1.85%
2-input NAND	4.5%	1.5%	1.45X	1.10X	1.68X	1.53X	1.30%
Iso Area INV	3%	1.5%	1.21X	1.10X	1.31X	1.53X	0%
Incr. Area INV	6%	13%	1.86X	3.88X	2.22X	7.04X	2.40%

four cases. This is due to the presence of eSiGe in PMOS which is a major contributor to the overall stress in the channel. As a result, for PMOS devices, altering Layout Property #1 (increasing the active area) results in the maximum improvement as compared to the improvement achieved by modifying the other two properties. However, in the case of NMOS devices, increasing active area results in pushing away the STI, whose contribution to the overall channel stress is relatively smaller. The longitudinal stress due to nitride is increased upon the alteration of Layout Property #2, and Layout Properties 2–3 are the major contributors to the drive current improvement in NMOS devices.

Table 4.2 summarizes the results of changing Layout Properties 1–3 in a few standard cells. It reports the percentage drive current improvement, leakage current increase, and the percentage increase in the output capacitance (assuming an FO4 output loading). It also reports the leakage current increase for identical drive current improvements through V_{th} reduction. Comparing the leakage current increase for stress-aware layout optimization to V_{th} reduction re-establishes the superiority of the stress-aware layout optimization. For a 3-input NOR gate, the PMOS leakage current increased by 4X when

the layout was optimized to exploit stress dependencies, while the corresponding increase for the V_{th} reduction case was 9.2X. The increase in NMOS leakage for a 3-input NAND gate was found to be 2X for stress-based layout optimization, and 2.4X for the case of V_{th} reduction. Application of Layout Property #1 increased the S/D capacitance since $L_{S/D}$ was increased, but, as shown in Table 4.2, this increase was very small (<3% if we assume an FO4 output loading).

In this same manner, we modified the layout properties from Section 4.5 in ~25 standard cells in a 65nm industrial library, creating a stress-enhanced version of each cell. For the majority of standard cells, the stress-enhanced versions are the same area as the original cells, thus, there is no area penalty. However, since there are no series/stacked devices in inverter layouts, there is negligible space to modify Layout Property #1. The capacitance increase for the "Iso Area INV" is 0% as reported in Table 4.2, because there is only space for the application of Layout Property #3, which does not affect capacitance. Therefore, we decided to create a second, slightly larger, stress-enhanced version of each inverter cell (with ~20% area increase per cell) that achieved larger drive currents (13% increase for PMOS and 6% increase for NMOS). Since the inverters, however, only make up a small subset of our standard cell library, the overall impact on circuit area is <0.5% (as shown later in Table 4.3). The final stress-enhanced standard cell library is comprised of different sized inverters (iso-area and increased-area versions) as well as 2- and 3-input NAND and NOR gates of varying strengths.

As mentioned in Section 4.4, the position of a device within a layout also affects its stress, and, therefore, its drive current. This position-dependent drive current enhancement can significantly hurt the performance of some circuits. This fact was verified using the

circuit shown in Figure 4.10, which contains the schematic and partial layout of a basic domino implementation of a 2-input OR gate. Keeper device P2 is a weak PMOS that is used to hold the high state at node N during the evaluation period of the clock, so that N is not discharged by the NMOS leakage currents. The keeper, P2, should be sized large enough to replace the NMOS leakage current and sustain a high voltage at N, but, at the same time, it should be small enough so that the pull-down network can discharge N quickly to minimize the short-circuit current.

Figure 4.10 shows two possible layout scenarios for the three PMOS transistors. In one case P2 is located between P1 and P3, while in the other case P1 is in the middle. As shown in Section 4.4, for the two scenarios the drive current for P2 differs by ~8%. This means that the first scenario has higher drive current for keeper P2 than the expected value. As the keeper fights against the pull-down stage, there is a performance loss. Hspice simulations show that the time taken to discharge node N increases by ~12%. This performance loss can worsen for more aggressively sized cases. For these Hspice

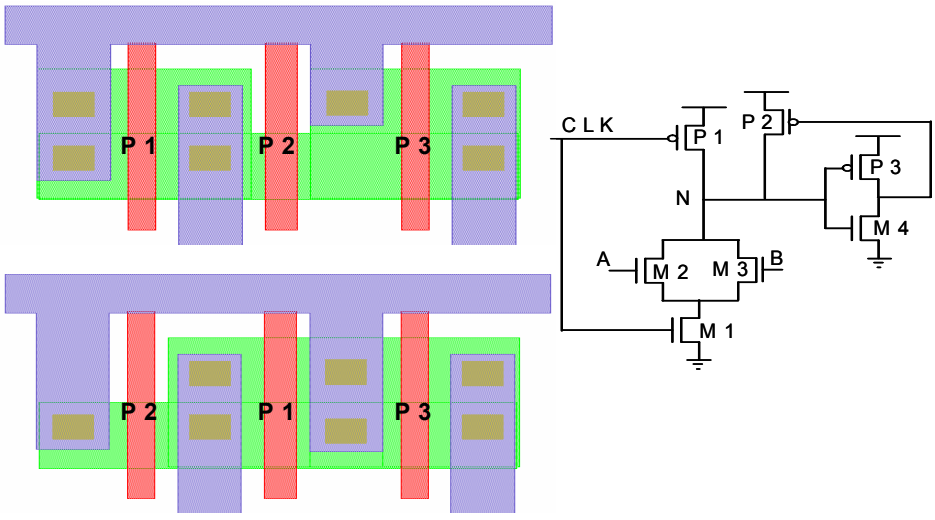


Figure 4.10. Basic Domino Gate and Two Possible Layouts for the PMOS Devices.

simulations, we approximated the drive current increase due to stress by changing the relevant mobility numbers in the transistor models.

4.7 Optimization Methodology

Stress-based performance enhancement provides a better leakage versus performance tradeoff than V_{th} assignment (as discussed previously in Section 4.3.2). However, when the standard cell area is fixed (i.e., the stress-enhanced version occupies the same/slightly higher amount of area as the original version), we can only obtain limited average drive current improvement through stress-aware layout optimization (<10%). Therefore, we combine stress-optimized assignment with dual-V_{th} assignment to simultaneously achieve a larger range of current improvement and more fine-grained control over the performance enhancement (and, consequently, the increase in leakage). Figure 4.11 shows the leakage and switching delays for various combinations of V_{th} and stress-based optimization for a 3-input NOR gate. Low stress (L_{stress}) optimization corresponds to a standard cell in the library that has not been optimized for stress enhancement (by altering the layout

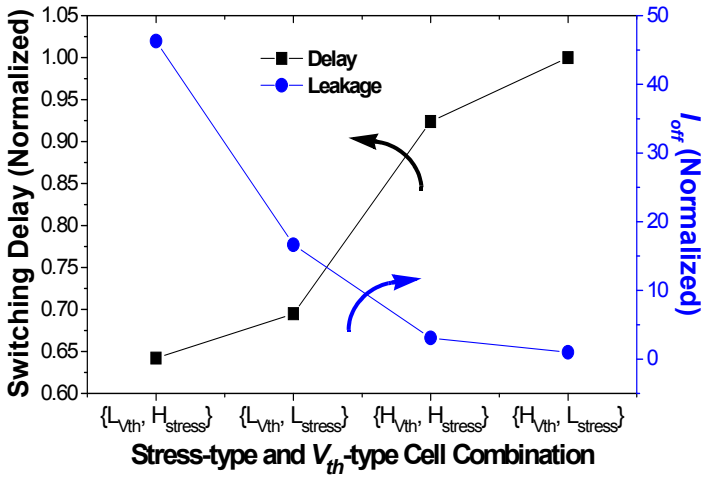


Figure 4.11. Leakage and Switching Delays for Various Combinations of V_{th} and Stress-based Optimization for 3-input NOR Gate.

properties), while high stress (H_{stress}) optimization corresponds to the layout optimized version of the standard cell. For the dual-V_{th} approach, a gate has only two options to choose from, high-V_{th} (H_{Vth}) or low-V_{th} (L_{Vth}). Introducing stress-based, layout-optimized cells provides an additional reduced leakage option (when performed on a high-V_{th} cell) for gates that require moderate improvements in performance, thereby saving leakage power. Additionally, it also provides a higher performance option when combined with low-V_{th} to further reduce delay.

For simultaneous V_{th}/stress optimization level selection and sizing optimization, we use an iterative approach similar to [22] that can be divided into two main parts:

1. A certain number of gates in each iteration are assigned to the low-V_{th} or high stress optimization level.
2. The circuit is then rebalanced by reducing the size of the affected gates and other gates are re-sized to compensate for the area reduction (the objective is iso-area).

Initially, all gates are set to their $\{H_{Vth},L_{stress}\}$ version, to maximize leakage savings. Then, in each iteration, a merit function is evaluated for all gates in a circuit. This merit function rates the increase in total leakage with respect to the performance gain of the circuit. Gates with the highest merit are selected first and set to the next highest performance level. The performance levels for our library are shown in the x-axis of Figure 4.11, and, from left to right, are ordered from highest performance (and leakage) to lowest performance (and leakage). This order holds for all standard cells in our library. The merit function is shown in (4–3):

$$\text{Merit}(G) = \frac{\Delta I_{off}(G)}{\Delta D(G)}$$
$$\text{where } \Delta D(G) = \sum_{\text{arcs}}^{\alpha} \Delta d_{\alpha}(G) \cdot \frac{1}{k + Slack_{min} - Slack_{\alpha}} \tag{4–3}$$

Here, $\Delta d_{\alpha}(G)$ is the impact that increased gate performance has on a particular timing arc, α; k is a small negative number; and $Slack_{min}$ is the worst slack seen in the circuit. This weighting function takes the value $1/k$ for timing arcs on the critical paths, and approaches zero for less critical timing arcs.

Once the merit function is evaluated, a circuit's gate sizes are no longer optimal since one or more gates have been assigned to a higher performance level. The resulting decrease in delay creates excess area which can be recovered from the now oversized gates. By shifting this excess area to undersized regions, we can improve performance without increasing area (or only increasing it by a small amount). The candidates for reduction include the modified gate itself along with any gates sharing a timing path with the modified gate. Because modifying a gate has a greater effect on nearby gates, we can identify a modified gate's core of influence to a predetermined logic depth based on the distance of gates (sharing a timing arc with the modified gate) from the changed gate. This depth was experimentally determined to be three levels of logic [22]. For the purpose of resizing, we use a delay-sensitivity-based sizing optimization algorithm [71]. The pseudo-code for a given value of target critical delay (T_T) is shown as Algorithm 4–1. Note that Lines 3 and 4 merely provide one set of initial values for T_C and T_N such that the conditions of the while loop are satisfied in the first iteration.

Algorithm 4–1 STRESS_OPT(T_T) // T_T = Target Delay

```
1:  Set all cells in netlist to {H_Vth,L_stress} version
2:  Run Initial STA and baseline sizing
3:  T_N = T_T + 1 // T_N = new critical path (CP) delay
4:  T_C = T_N + γ + 1 // T_C = current CP delay
5:  // γ = small constant, checks for >minimal changes in T_C
6:  while ( (T_N > T_T) and ((T_C - T_N) > γ) )
7:      T_C = T_N
8:      Evaluate Merit(G) for all gates, G // see (4–3)
9:      Move gates with highest Merit(G) to next highest performance level
10:     Rebalance circuit through sizing
11:     Update STA, find new critical delay, T_N
12: end while
```

The next section discusses the experimental results obtained when applying this optimization algorithm to 12 different benchmark circuits.

4.8 Experimental Setup and Results

The following section describes the library characterization used within our experimental setup, as well as the results obtained from using the proposed optimization scheme on a number of benchmark circuits.

4.8.1 Library Characterization

To implement our optimization methodology, we first had to characterize our stress-enhanced standard cell library and determine the decrease/increase in propagation-delay/leakage-power, respectively, that the standard cells achieved while exploiting the layout dependencies of stress. The characterization flow is illustrated in Figure 4.12 and captures the relative change in propagation delay and leakage power, as compared to the "unstressed" version of a particular standard cell. While characterizing one standard cell,

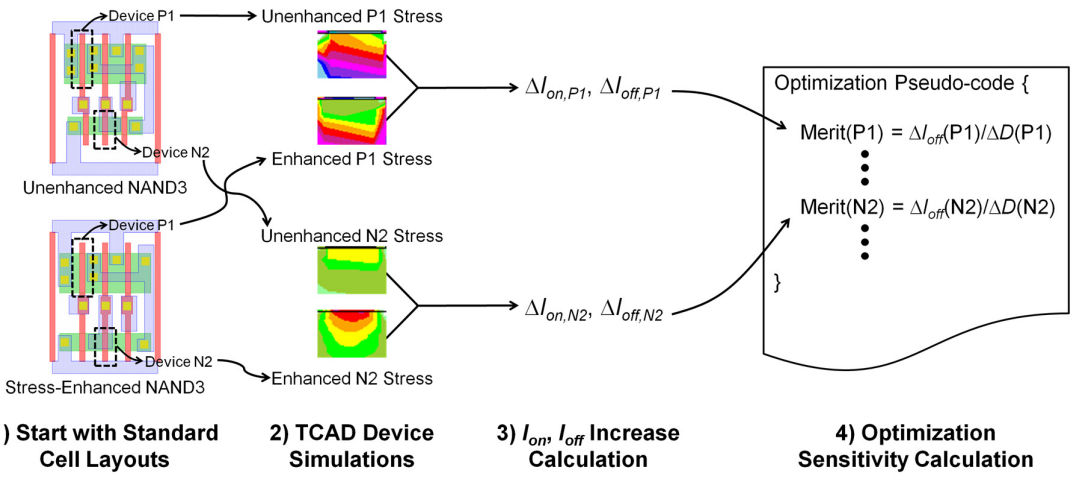


Figure 4.12. Custom Library Characterization Flow for Stress-aware Optimization.

we simulated both the stress-enhanced version and its unstressed counterpart in *Tsuprem4* and *Davinci*, as discussed in Section 4.4. From these simulations, we were able to calculate the relative increase in I_{on} and I_{off} (referred to as $\Delta I_{on}(X)$ and $\Delta I_{off}(X)$, respectively) for each device, X, within the standard cell. These $\Delta I_{on}(X)$ and $\Delta I_{off}(X)$ values for every PMOS and NMOS device (in every standard cell in our library) were then input directly into the optimization engine. Within the optimization algorithm, $\Delta I_{on}(X)$ is translated to decreasing propagation delay by using an inverse relationship fit: $\Delta d_{\alpha}(X) \propto \frac{1}{\Delta I_{on}(X)}$. Finally, these values, $\Delta d_{\alpha}(X)$ and $\Delta I_{off}(X)$, are used directly in the merit function described in (4–3).

In order to examine the effect that neighboring cells had on the channel stress of a device, we conducted a simple experiment where the value of I_{on} for a minimum-sized inverter in isolation was compared to the same minimum-sized inverter which had inverters as neighbors on both sides (representing a more "dense" context). We chose the minimum-sized inverter because of all of the standard cells, it was the most sensitive to

changes in context. For the stress-enhanced inverter cell, we observed a 0.8% higher I_{on} and a 2.0% higher I_{off} in the case where neighboring cells were included. However, the corresponding gains in I_{on} and I_{off} (ΔI_{on} and ΔI_{off}) for the stress-enhanced version (compared to the unoptimized version) decreased by <0.1% and <1%, respectively, while considering neighbors. Since the I_{on}/I_{off} gains achieved for stress-enhanced layouts showed little sensitivity to changes in context and because circuit level TCAD simulations were not possible (due to runtime and convergence issues), we used the library characterization of isolated cells to drive the circuit-level analysis in this chapter. In the proposed circuit-level optimization (discussed in Section 4.7), critical cells are iteratively exchanged with their stress-enhanced (or dual-V_{th}) counterparts. While considering the optimization of one particular cell within one iteration, only the type of enhancement is modified. All other parameters like neighborhood, size, and cell type (NAND, NOR, etc.) are held constant. Since the merit function described in (4–3) is dependent on ΔI_{on} (which determines Δd_{α}) and ΔI_{off}, the accuracy of our optimization technique is dependent on the sensitivity of the I_{on}/I_{off} gains to changes in context. As mentioned previously, we found that ΔI_{on} (ΔI_{off}) changed by <0.1% (<1%) when context was varied from isolated to dense. Therefore, the proposed library characterization of isolated cells is accurate and can be used within our merit-based optimization scheme, independent of context.

4.8.2 Experimental Results

The algorithm described in Section 4.7 was implemented in C and tested on ISCAS85 benchmark circuits, two DSP circuit implementations ("Viterbi1" and "Viterbi2"), and a USB 2.0 controller implementation. The benchmarks vary in size from 166 to 37560

gates. The circuits were synthesized using an industrial 65nm CMOS technology with the following specifications:[3]

- $V_{DD,nominal}$ = 1V
- HVT, NMOS V_{th} = 334mV
- HVT, PMOS V_{th} = -391mV
- LVT, NMOS V_{th} = 243mV
- LVT, PMOS V_{th} = -280mV

The resulting spread in I_{on} and I_{off} (between HVT and LVT) was 1.24X/1.32X and 16X/29X, respectively, for NMOS/PMOS transistors. All of the standard cells (both the original and the stress-enhanced versions) in our library were characterized (using Hspice) at both the high- and low-V_{th} values. The layout-dependent characteristics (e.g., rise/fall delay, rise/fall power, etc.) and parasitics (such as junction capacitance and S/D resistance) for each cell were captured during the Hspice characterization. All of the improvements discussed in this section use a dual-V_{th} optimization (using simultaneous V_{th} selection and gate sizing) as the basis for comparison.

Figure 4.13 shows the leakage power versus critical delay curves for the two techniques: dual-V_{th} assignment and dual-V_{th} assignment combined with stress-aware layout optimization, for one of the larger circuits, c7552. As mentioned earlier, combining stress-based layout optimization with V_{th} assignment provides a better range and more

[3] Reported V_{th} values were obtained using the industry standard "constant current method" [72], where V_{th} is determined by extracting V_{GS} at the point where $|I_{DS}| = 100\text{nA} \cdot \frac{W}{L}$ (with $V_{DS} = V_{DD,nominal}$).

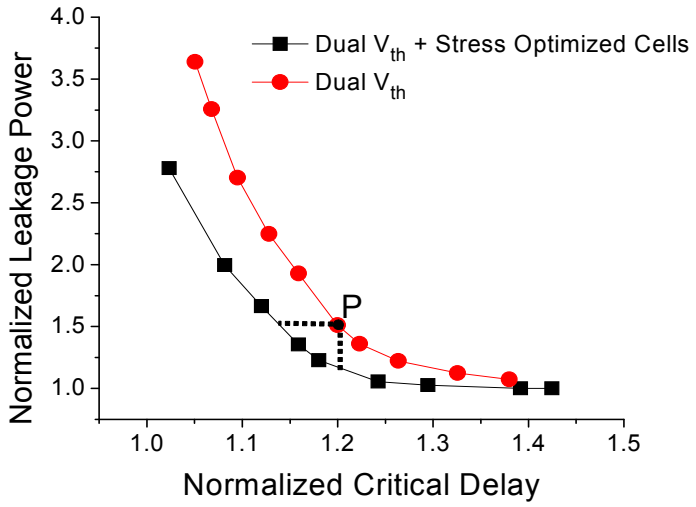


Figure 4.13. P_{leak} vs. Delay for Dual-V_{th} and Proposed Approach for Benchmark c7552.

fine-grained control of performance enhancement as compared to the dual-V_{th} based assignment (see Table 4.3 for the cell combinations used in each optimization scheme). This is clearly seen in Figure 4.13 while comparing both the critical delay for the two techniques at the same value of leakage (iso-leakage), as well as the leakage power at the same value of critical delay (iso-delay). The key metric that we use in our comparisons is known as hardware intensity (η), which was proposed in [73] for quantifying the tradeoff between power and delay of a design. A hardware intensity of x means that a 1% decrease in delay leads to an x% increase in power. The hardware intensity for the majority of blocks in a microprocessor design is between 2 and 3 [74]. Thus, for a fair evaluation of the proposed approach, we present results for points on the power-delay curve that correspond to a hardware intensity value between 2 and 3. One such point is shown as "P" in the leakage-power-delay tradeoff curve (η = 2) in Figure 4.13. For the circuit, c7552, our proposed optimization results in 22% lower leakage power for iso-delay, and 5.4% lower delay for iso-leakage, when compared to dual-V_{th} based assignment at point P.

Table 4.3. Stress and V_{th} Combinations.

		Cell Combinations
(1)	**Combined stress-enhancement and dual-V_{th}**	$\{L_{Vth}, H_{stress}\}$, $\{L_{Vth}, L_{stress}\}$, $\{H_{Vth}, H_{stress}\}$, $\{H_{Vth}, L_{stress}\}$
(2)	**Only dual-V_{th}**	$\{L_{Vth}, L_{stress}\}$, $\{H_{Vth}, L_{stress}\}$
(3)	**Only stress-enhancement**	$\{H_{Vth}, H_{stress}\}$, $\{H_{Vth}, L_{stress}\}$

Figure 4.14 shows how the percentage improvement (of our combined method over dual-V_{th}) in leakage power and critical delay, as well as the corresponding area overhead varies with hardware intensity for c7552. Percentage improvement in leakage power increases with increasing hardware intensity because the leakage-power-delay curves for our approach and dual-V_{th} assignment move further apart as delay decreases (or hardware intensity increases). The improvement in critical delay also increases with increasing hardware intensity. The area overhead, however, shows an initial increase as more gates require higher performance, but then becomes fairly constant at higher values of hardware intensity. For the remainder of this section, we report power and delay improvement numbers for points on the leakage-power-delay curves that correspond to a hardware intensity of 2.

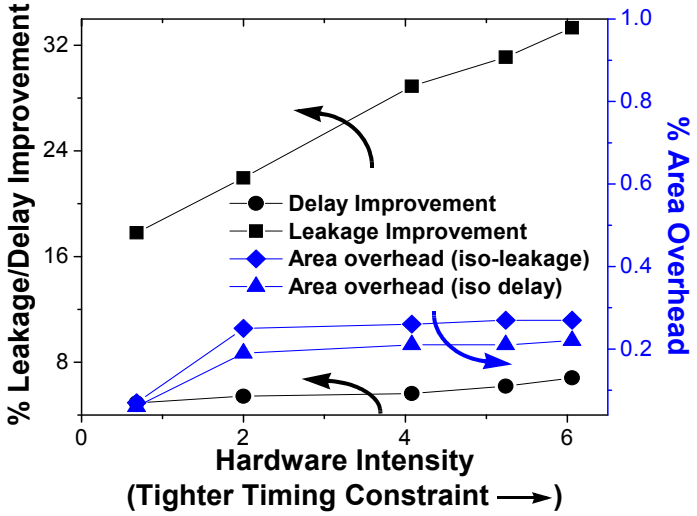


Figure 4.14. Delay, P_{leak}, and Area Overhead vs. Hardware Intensity.

Table 4.4. Improvement in Leakage and Delay Compared to Dual-V_{th} based Assignment.

Circuit	Number of gates	Comparison for iso-delay against only dual-V_{th} assignment				Comparison for iso-leakage against only dual-V_{th} assignment			
		Stress + V_{th} based assignment		Only Stress based assignment		Stress + V_{th} based assignment		Only Stress-based assignment	
		Improvement in leakage	Area overhead	Improvement in leakage	Area overhead	Improvement in delay	Area overhead	Improvement in delay	Area overhead
c432	166	38.5%	0.3%	5.4%	0.5%	5.0%	0.5%	3.6%	0.6%
c499	962	20.4%	0.9%	5.1%	0.9%	4.6%	0.9%	3.4%	1.0%
c880	390	33.7%	0.1%	12%	0.2%	5.8%	0.3%	2.3%	0.3%
c1908	432	22.5%	0.6%	7.4%	0.7%	4.7%	0.9%	3.0%	0.9%
c2670	964	14.7%	0.1%	5.1%	0.2%	5.2%	0.3%	3.6%	0.3%
c3540	962	23.9%	0.2%	4.7%	0.3%	4.7%	0.3%	2.5%	0.3%
c5315	1750	22.9%	0.2%	4.9%	0.3%	4.9%	0.2%	2.6%	0.2%
c6288	2470	20.1%	0.9%	5.9%	0.9%	4.6%	0.9%	3.0%	0.9%
c7552	1993	22.0%	0.3%	4.8%	0.2%	5.4%	0.2%	3.1%	0.3%
Viterbi1	14503	21.5%	0.3%	4.9%	0.4%	5.3%	0.3%	2.9%	0.5%
Viterbi2	34082	22.6%	0.3%	5.1%	0.4%	5.2%	0.2%	2.7%	0.4%
USB	37560	22.4%	0.3%	5.2%	0.3%	5.2%	0.4%	2.8%	0.3%
Average		**23.8%**	**0.4%**	**5.9%**	**0.4%**	**5.1%**	**0.5%**	**3.0%**	**0.5%**

Table 4.4 summarizes the improvements seen in two comparisons: combined stress-enhancement and dual-V_{th} (which uses the cell combinations shown in (1) in Table 4.3) versus only dual-V_{th} (see (2) in Table 4.3); and stress-enhancement (see (3) in Table 4.3) versus only dual-V_{th}. The first two columns state the name of the test circuit and its size. The next four columns report the percentage improvement in leakage over the dual-V_{th} case and the corresponding area overhead for iso-delay (for both comparisons). The last four columns show the percentage improvement in critical delay and the corresponding area overhead for iso-leakage-power (for both comparisons). The small value of area overhead occurs because of the increased area variants of the layout-optimized inverter cells (mentioned in Section 4.6).

The results clearly show that our combined approach significantly improves the leakage power for iso-delay, and also improves critical delay for iso-leakage, when compared to dual-V_{th} based assignment. We get up to a 38.5% (23.8% on average) improvement in leakage for iso-delay, and up to a 5.8% (5.1% on average) improvement in delay for iso-leakage. The area overhead is very small for both the cases – less than 0.5% on average across all 12 circuits. It is worth noting that while our delay improvements are similar to those published in [36], our proposed technique provides the 5.1% delay improvement (on average) for iso-leakage.

As mentioned previously, Table 4.4 also includes a one-to-one comparison of stress-enhancement versus dual-V_{th}, where stress-enhancement achieves up to a 7.4% (5.9% on average) improvement in leakage for iso-delay, and up to a 3.6% (3% on average) improvement in delay for iso-leakage (compared to dual-V_{th}). The discrepancy between the leakage improvement of the combined approach (stress + dual-V_{th}) versus dual-V_{th} (23.8% on average) compared to only stress-enhancement versus dual-V_{th} (5.9% on average) arises because the point on the stress-enhancement leakage/delay curve where hardware intensity equals 2 ($\eta = 2$) occurs at a larger delay (e.g., a point to the right of P in Figure 4.13). This is explained by the fact that stress-enhancement alone can only achieve $< 1/2$ of the performance enhancement of dual-V_{th}. Thus, the leakage comparison between stress-enhancement and dual-V_{th} occurs in the region of leakage-versus-delay where stress does not have as large of an advantage over dual-V_{th} (note the smaller gap between the two curves in Figure 4.13 as you move towards larger delays). However, at the new comparison point, for this framework and technology, stress-enhancement still

outperforms dual-V_{th} both in leakage optimization as well as delay optimization. This is noteworthy because using stress-enhancement by itself eliminates the extra masks and processing steps required by dual-V_{th} designs, which reduces process complexity and cost. Furthermore, the stress-enhancement versus dual-V_{th} improvement numbers are limited by the fact that we require small or no area overhead for the redesigned standard cells. Using more advanced techniques, we could further improve the stress-enhanced tradeoff between area and performance, which will increase the performance gap between stress-enhancement and dual-V_{th}.

Figure 4.15 shows the percentage of gates assigned to low-V_{th} for the dual-V_{th} assignment, as well as the combined "stress enhancement + dual-V_{th}" approach. These numbers are reported for iso-delay points on the leakage-delay curves corresponding to a hardware intensity of 2. As expected, for the combined approach, a lesser number of gates are assigned to low-V_{th} as compared to dual-V_{th} assignment. This is because for the dual-V_{th} assignment, not all gates assigned to low-V_{th} need such a large performance improvement. Combining stress-optimized cell assignment with dual-V_{th} assignment

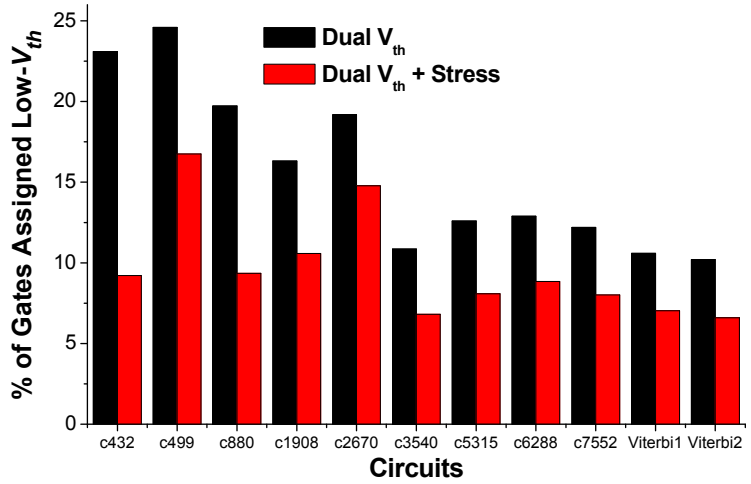


Figure 4.15. Percentage of Low-V_{th} Gates used in the Dual-V_{th} and Proposed Approach.

provides an additional lower leakage option for the cells that require moderate improvements. This reduces the number of cells that are assigned to low-V_{th}, which, in turn, results in lower leakage current. Typically, the number of gates assigned to low-V_{th} for the combined approach is ~35% lower than the number for dual-V_{th} assignment.

To further investigate the tradeoff that exists between leakage power savings and area overhead, we performed another experiment using a richer library comprised of higher area, stress-enhanced versions of all the cells. The area overhead for the higher area versions was ~20% per cell, and every cell in the richer library had three variants: an original unoptimized version; an iso-area, stress-enhanced version; and an increased area, stress-enhanced version. The richer library provided more intermediate, low-leakage options (in addition to the low-V_{th} cell) for gates requiring moderate improvements. By providing these intermediate performance alternatives, the overall leakage power (for iso-delay) is further reduced as compared to dual-V_{th} assignment. Figure 4.16 shows the comparison between the "stress-enhancement + dual-V_{th} assignment" optimization for the richer library and the original, stress-optimized library (with increased area versions for

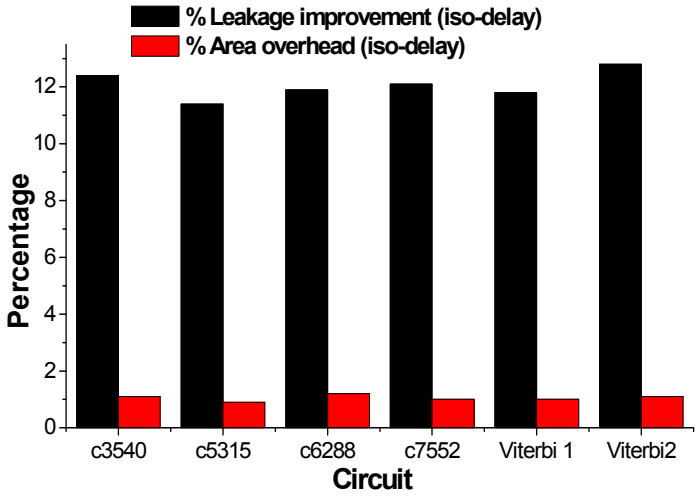


Figure 4.16. P_{leak} Improvement and Area Overhead for the Richer Library vs. Original.

inverters only). It plots the leakage power improvement (for iso-delay) and the corresponding area overhead obtained by using the richer library (compared to the original stress-enhanced library) for six of the larger circuits. On average, using the richer library further improved the leakage power (at iso-delay) by ~12% for an area overhead of ~1% over joint assignment using the original library. This experiment shows that there is scope for further improvement using the richer library. However, the richer library also incurs a higher characterization cost due to the large number of variants for each cell. One approach to minimize this cost would be to only create multiple versions of cells that are used most often (typically the smaller gates such as inverters, NAND's, NOR's, etc.).

4.9 Summary

In this chapter, we explored the modification of standard cell layouts in order to optimize the stress-based performance enhancement, and proposed a block-based optimization algorithm that combined stress-enhancement with dual-V_{th} assignment to achieve performance gains in leakage or delay. We studied the dependence of drive current improvement on layout parameters like source/drain length and contact placement, and found that the performance of any given layout could be enhanced by increasing the active area length. Based on our observations, we exploited a set of layout properties which maximized the performance improvement of a standard cell without increasing area. When these properties were modified in standard cells from a 65nm industrial library, PMOS and NMOS drive currents attained an average performance enhancement of 6% and 4.4%, respectively, without increasing the cell area. The corresponding average increase in leakage was found to be 2.2X and 1.5X for PMOS and NMOS devices,

respectively. Next, we combined the assignment of these stress-optimized cells with V_{th} assignment in order to optimally tradeoff leakage power and performance. When compared to the traditional dual-V_{th} based assignment technique, the new approach reduced leakage current by 23.8% on average for identical delay, and improved critical delay by 5.1% on average for identical leakage, with a very small area overhead (<0.5%).

CHAPTER 5

STEEL: A TECHNIQUE FOR STRESS-ENHANCED STANDARD CELL LIBRARY DESIGN

As discussed in Chapter 4, three of the four main mechanical stress sources in today's processes – STI, nitride, and eSiGe – are all dependent on common layout parameters in modern standard cells. The two most dominant layout properties that affect mechanical stress and are customizable within standard cell design are source/drain (S/D) active area and contact placement. Larger S/D areas allow for greater amounts of eSiGe (in PMOS devices) and nitride (in both types of devices), which enhances mechanical stress in the channel. Contact placement, however, disrupts the continuity of the nitride layer and, consequently, lowers the contribution of the nitride layer to channel stress. Hence, contacts placed farther away from the channel will increase the amount of nitride adjacent to the channel, enhancing channel stress. Overall, the layout dependencies of stress are well documented [29,33,58], but little research has been dedicated to developing new standard cell library design techniques that exploit these dependencies.

Thus, in this chapter we propose a new standard cell design methodology that strives to fully exploit the layout dependencies of mechanical stress. Our library design methodology differs from previous mechanical stress work in that it employs a cell-level, library-wide enhancement technique that not only increases within-cell stress, but also

increases cell-to-cell stress. Since most standard cells in a typical library have source/drain V_{DD} and V_{SS} ties adjacent to one or both edges of the cell, our new, stress-enhanced libraries share these ties across cell placement and route boundaries as illustrated in Figure 5.1. By sharing the V_{DD} and V_{SS} nodes, stress is enhanced in both the edge devices as well as their neighbors, increasing I_{on} and I_{off} by up to ~20% and ~3.5X, respectively for PMOS devices, and 7.5% and ~2X, respectively for NMOS devices.

The remainder of the chapter is organized as follows. Section 5.1 describes the technique used in our proposed standard cell design methodology. Section 5.2 presents our standard cell design and its ease of integration within state-of-the-art VLSI design flows. Finally, Section 5.3 discusses our results and Section 5.4 concludes the chapter with a brief summary.

5.1 A Technique for Enhancing Stress in Standard Cell Layouts

As stated in Chapter 4, mechanical stress in MOSFET channels depends on a number of layout parameters. However, the amount of mechanical stress in a typical CMOS device

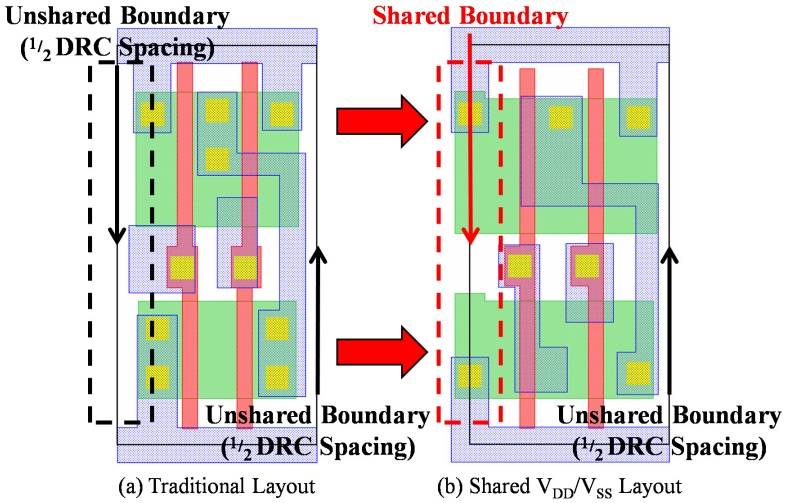


Figure 5.1. Traditional Standard Cell Layout vs. Proposed Shared Source/Drain Layout for a 2-input NAND.

is not only a function of its own layout parameters (S/D area, contact placement, etc.), but also of its neighbors' parameters. Thus, NMOS and PMOS devices that share their S/D regions with other transistors have significantly higher channel stress (and, hence, drive current enhancement) than those at the edges of an active region (which are therefore bordered by STI), even for identical active area length and contact placement. For NMOS devices, this is mainly due to the fact that STI has a negative impact on the amount of tensile stress induced in the longitudinal direction, resulting in lower values of tensile stress in edge devices compared to devices towards the center. For PMOS devices, stress due to STI enhances channel stress, however, since eSiGe has a much stronger contribution than STI, "center" PMOS devices also exhibit considerably higher channel stress as they are surrounded by more eSiGe. Therefore, in the presence of mechanical stress, two devices with identical layout parameters (W, L, $L_{S/D}$, contact placement, etc.) may differ significantly in drive current, depending upon their positions in the layout (even when neglecting process variation).

From a standard cell design perspective, one would ideally avoid these stress-based variations and move to a more uniformly stressed standard cell to minimize context dependencies and performance uncertainty. By sharing the V_{DD} and V_{SS} source/drain ties across standard cell boundaries, we can effectively increase the number of "center" devices (devices with at least one other transistor on both sides) in a given standard cell. This results in higher channel stress in the devices of such cells, since all of the affected devices will have more neighbors (which means more eSiGe, smaller STI regions, more nitride, etc.). Figures 5.2 (a) and (b) illustrate our shared V_{DD} and V_{SS} source/drain connection technique (referred to as the STEEL – **ST**r**E**ss **E**nhanced **L**ibrary – technique

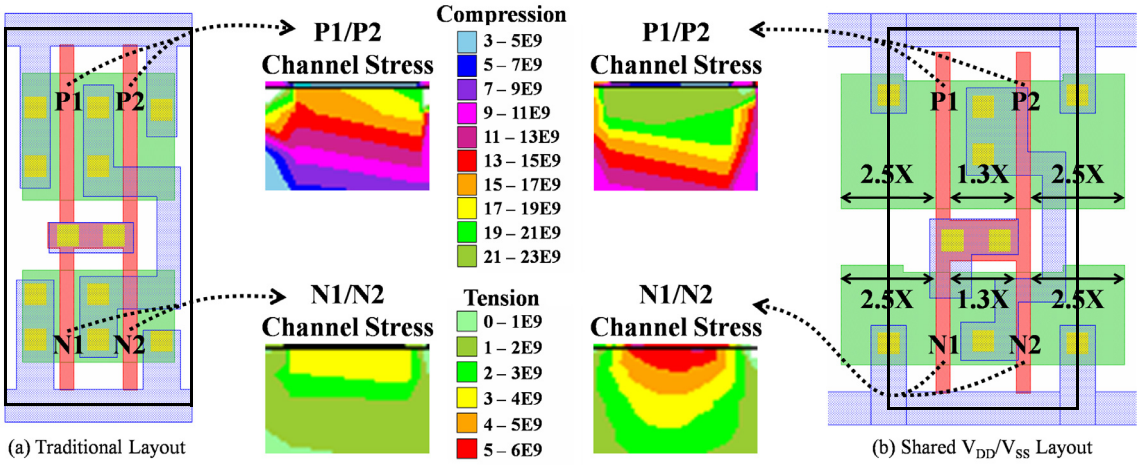


Figure 5.2. Impact of Shared V_{DD}/V_{SS} Approach on Stress (Pa) in a Two-Finger Inverter. (Note: The channel stress in N1 (P1) is identical to N2 (P2) due to symmetry)

for the remainder of the chapter). Figure 5.2 (a) depicts the traditional standard cell layout (for an inverter with two fingers) where the active area edge is placed at a location >=1/2 the design rule space from the standard cell boundary (the black rectangle that encapsulates the cell). However, since most standard cells in a typical library have at least one cell edge that is adjacent to a V_{DD} and V_{SS} S/D, we can share the connection between cells, effectively doubling the S/D active area and eliminating STI between the two cells. The edge devices achieve the largest increase using this approach – typically $L_{S/D}$ increases by >2X – and their induced channel stress now becomes more comparable to the stress in the "center" devices. Therefore, sharing the V_{DD} and V_{SS} connections between standard cells will not only lead to a more uniform distribution of channel stress, but will also improve the overall drive current of the standard cells (shown in the channel stress contour plots in the center of Figure 5.2). The actual "sharing" occurs in Figure 5.2 (b) where the Metal-1 connections from V_{DD} and V_{SS} have been moved to the cell boundary. In this case, PMOS and NMOS drive currents increase by 13.5% and 6.3%, respectively, while leakage current increases by 2.8X and 1.6X. Furthermore, one of the strengths of

STEEL is that it achieves these improvements in stress uniformity and drive current with no cell area increase (i.e., the area encapsulated by the black place and route boundaries in Figure 5.2 is identical for both cells (a) and (b)).

5.2 Implementation of STEEL in Standard Cell Design

In order to develop a 65nm STEEL standard cell library that accurately captured stress effects and ensured compatibility within existing VLSI design tools (e.g., synthesis tools, place and route tools, etc.), we created a design flow which is described below and illustrated in Figure 5.3. This design flow is executed on a cell-by-cell basis, and begins by capturing the effects of stress for each device within a cell. We use *Tsuprem4* to simulate the fabrication steps and *Davinci* 3D TCAD to capture the stress-enhanced device parameters. Then, we calibrate our TCAD model with an Hspice model and extract the effects of stress into one device-specific multiplication factor: the low-field mobility multiplier ($\mu_{0,STRESS\_MULT}$). This modified Hspice model is then used within Cadence's *Signalstorm* (a library characterization tool) to calculate the propagation delays and power consumption for a given cell, which is eventually output in Synopsys's LIBERTY file

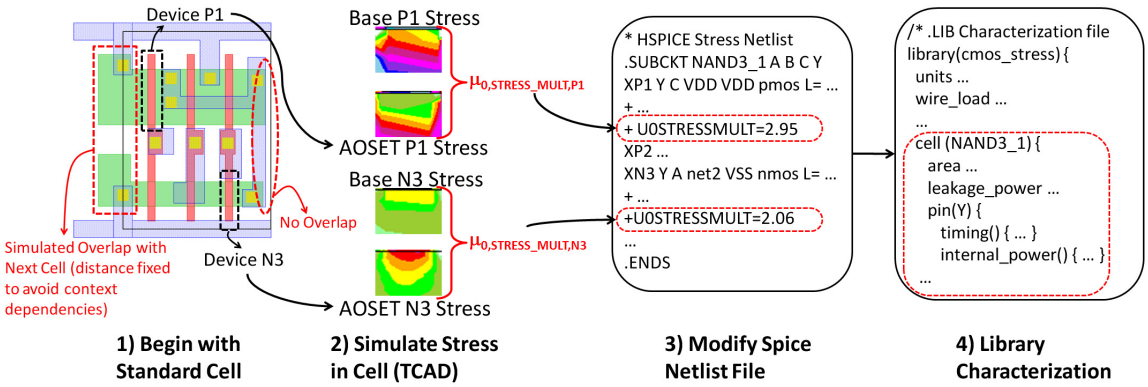


Figure 5.3. STEEL Library Characterization Flow.

format. This LIBERTY file can be used in a number of industry standard synthesis and/or automated place and route (APR) tools.

The remainder of this section describes the STEEL standard cell design flow in more detail and concludes by describing common issues encountered and how they were resolved. We implemented our design flow on a reduced set of the most commonly used standard cells – 33 standard cells in total.

5.2.1 *Tsuprem4* and *Davinci* Device Simulation

Our design flow begins by using *Tsuprem4* to simulate the fabrication of a particular device and capture the process-induced stress. *Davinci* 3D TCAD tool is then used to capture device behavior under stress by solving for stress-based mobility enhancement equations. We used a TCAD device simulator for this work because currently, to our knowledge, there are no industry-standard device models that capture all of the layout-dependent effects of stress. BSIM4 captures only the STI-related stress impact on effective mobility (μ_{eff}), saturation velocity (v_{sat}), and threshold voltage (V_{th}). However, Chapter 4 showed that other layout parameters also play a critical role in determining the amount of mechanical stress induced in a channel. Therefore, to capture these effects we simulate each standard cell in *Tsuprem4* and *Davinci*, and extract the new, stress-enhanced low-field mobility (μ_0) at $V_{GS} = V_{DD} = 1$V and $V_{DS} = 50$mV. By comparing a device's stress-enhanced mobility to its mobility without stress (the same TCAD simulation with the stress-analysis disabled), we can determine a device-specific scalar multiplier for μ_0: $\mu_{0,STRESS\_MULT}$. This multiplier is then used in our BSIM4 Hspice model, described next.

5.2.2 Stress-Enhanced BSIM4 Hspice Model

After calibrating *Davinci* device simulations to 65nm industrial Hspice models (by matching I_{on} and I_{off}), we adjust the BSIM4 model so that the low-field mobility multiplier, $\mu_{0,STRESS\_MULT}$, is included as a possible input parameter for both PMOS and NMOS devices. We simply scale the old value of μ_0 by the multiplier: $\mu_0 = \mu_{0,OLD} \cdot \mu_{0,STRESS\_MULT}$. Simultaneously, since our *Davinci* models already capture all of the sources of mechanical stress, we temporarily turn off the BSIM4 stress models for μ_{eff}, v_{sat}, and V_{th} by setting the stress effect parameters for mobility degradation/enhancement (*KU0)*, saturation velocity degradation/enhancement (*KVSAT*), and threshold voltage shift (*KVTH0*) to zero. The resulting *I-V* fit for minimum-sized NMOS and PMOS devices is shown in Figure 5.4, which verifies the accuracy of our model. For example, in these minimum-sized devices we find that our modified Hspice device models incur an average root mean square error in saturation current of ~3μA and ~0.7μA for the NMOS and PMOS devices, respectively. These Hspice device models eventually serve as the basis of our standard cell library characterization.

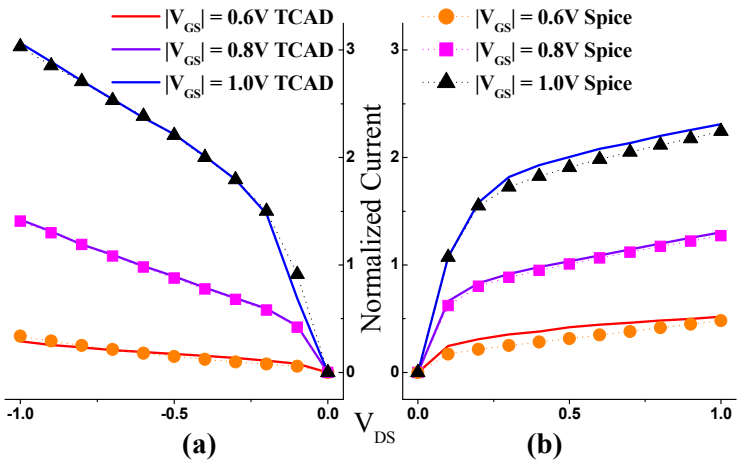


Figure 5.4. *Davinci* vs. Hspice *I-V* plots.
(a) PMOS (b) NMOS

5.2.3 Standard Cell Library Characterization

To make our new standard cell library compatible with existing digital, integrated circuit (IC) design flows, it is essential to be able to characterize the new standard cells and determine typical gate level parameters such as pin capacitance, propagation delay, dynamic and leakage power consumption, etc. To achieve this, we input our modified Hspice models into Cadence's *Signalstorm* delay calculator. *Signalstorm* then simulates our stress-enhanced gates over a number of output-loading and input-slew combinations and finally generates a LIBERTY characterization file. The LIBERTY file generation is the last step in the STEEL standard cell design flow and it enables the use of these new libraries within synthesis and APR tools with minimum additional overhead (described in more detail in Section 5.3.1).

5.2.4 Implementation Decisions in STEEL

There were several design decisions that needed to be resolved while creating a STEEL standard cell library. The first decision addressed the number of variants that could exist at an abutted boundary. These variants occur because many of the standard cells in a typical library cannot share the V_{DD} and V_{SS} connections at both edges of the cell. Instead, the adjacent S/D node is connected to some other net (e.g., the output node in a minimum-sized Inverter or NAND gate). For instance, refer to the 2-input NAND layout in Figure 5.1 (b). The NMOS drain on the right-hand side is tied to the output, Y. Therefore, this drain cannot be shared at the boundary with any arbitrary cell in a design whose left NMOS S/D is not connected to the same net. In this case, the PMOS source tied to V_{DD} could be shared, but only with a cell that has the same configuration (shared

PMOS, unshared NMOS) or a custom "Filler" cell designed for the "shared PMOS, unshared NMOS" case. Therefore, to keep the number of edge variants small, we implemented two types of standard cell edges: shared or unshared. If either the NMOS or PMOS S/D is not connected to V_{SS}/V_{DD}, respectively, then that edge of the cell is designed to be completely unshared. STEEL consequently has three different types of cells:

- Cells with both edges "shared" (such as the one in Figure 5.2 (b)).
- Cells with one "shared" edge and one "unshared" edge (previously discussed and illustrated in Figure 5.1 (b)).
- Cells with both edges "unshared" (similar to the layout shown in Figure 5.1 (a)).

Each standard cell in the library corresponds to only 1 of these 3 types, with the exception of inverters and buffers. To ease APR we designed two versions of inverter and buffer cells, one with the maximum number of shared connections and one with zero shared connections (both edges "unshared"). The "unshared" inverter and buffer cells reduce the placement/routing complexity involved during buffer insertion. For additional details of using STEEL libraries within APR, refer to Section 5.3.1.

The second design decision made was that a cell edge of a certain type (either shared or unshared) could only be abutted with an edge of the same type. In our implementation, we chose to let the APR tool handle this by passing it an additional set of constraints:

- Only abut "shared" edges with "shared" edges.
- Only abut "unshared" edges with "unshared" edges.

Details regarding the additional overhead needed to use STEEL within APR is included in Section 5.3.1.

The final implementation detail is a by-product of the layout dependency of stress. Since we are essentially extending the active area between standard cells, differing amounts of active overlap for different combinations of cells could significantly change the I_{on} and I_{off} currents for a given device. Therefore, context dependencies could easily arise if the STEEL library is not carefully designed. To illustrate this problem, consider the example in Figure 5.5, which shows two overlap cases for transistor, T_1. In the first case, the standard cell containing T_1 is placed next to a cell whose nearest device is T_2. The distance, X_{12}, between these two transistors corresponds to the active area length, $L_{S/D}$, of this source/drain region and directly affects the amount of stress induced in both T_1 and T_2. However, in the same design, the same cell type that contains T_1 is used again, but this time is placed next to T_3 and the S/D length increases by 1.3X. In this simple example, this 30% change will increase the drive current by ~10% (if we assume T_1, T_2, and T_3 are PMOS devices), which is substantial.

One way to handle this context dependency is to characterize the particular device, T_1 for every possible $X_{1,N}$ that could exist by abutting it next to any other "shared" edge in

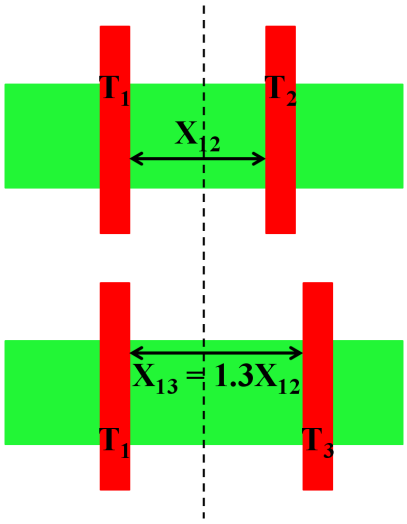


Figure 5.5. Context Dependency within STEEL Designs.

the library. However, since an industrial library typically has many hundreds of cells, this leads to an infeasibly large number of characterizations. Instead, we chose to fix the distance $X_{M,N}$, such that each device T_M and T_N are placed $0.5X_{M,N}$ away from the boundary. We selected a value for $X_{M,N}$ that achieved ~20% and ~8% increases in PMOS and NMOS I_{on} (for the edge devices) and increased I_{off} by ~4X and ~2X, respectively.

5.3 Experimental Results

In order to determine the strengths of the STEEL design methodology, we compared it to two industry design flows: single-V_{th} (using regular-V_{th}, or RVT, cells) and dual-V_{th} (using both RVT and low-V_{th}, or LVT, cells). These comparisons are included in Sections 5.3.2 and 5.3.3, respectively. We also describe a simple assignment technique in Section 5.3.4 which only applies the advantages of STEEL to critical cells, improving leakage at slower delay points or in unbalanced circuits. However, before we examine our results, we begin by briefly discussing how our place and route tools were configured to handle the STEEL library.

5.3.1 APR using STEEL Libraries

As mentioned previously in Section 5.2.4, the various standard cell edge types (either "shared" or "unshared" in our implementation) in the STEEL library add a small amount of complexity to cell placement. To minimize this complexity, we enforced a few additional constraints within the APR tool (discussed in Section 5.2.4). We accomplished this through a custom Tool Command Language (TCL) script that was designed and run within Cadence's APR tool, *Encounter*. Essentially, the script steps through each placed

standard cell in the design, starting with the top, left most cell, and continues from left to right across a single core row before proceeding to the next row down. As the script traverses the standard cell row (from left to right), it checks the adjacent cell edges. If the edges match, the TCL script moves to the next cell. However, if the edges do not match, the script checks if the opposite side of the right cell matches the current cell edge. If it does, the script flips the cell and continues. If neither sides match, then a filler cell is placed in between the cells, to ensure that design rules are satisfied. The penalty incurred is typically minimal, and we found that even with row utilizations of up to ~85%, the STEEL library can be placed and routed using the same floorplan and dimensions as the traditional standard cell libraries.

5.3.2 STEEL versus Regular-V_{th} Results

We begin our analysis by comparing the area, leakage power, and delay of STEEL designs to their traditional, single-V_{th}-based equivalent. The basis of our comparison was an industrial 65nm RVT library. Both libraries were characterized using the stress-enhancement models and flow described in Section 5.2 and pictured in Figure 5.3. With the new LIBERTY files, we were able to synthesize and place and route a variety of benchmarks using both libraries. In total, we implemented the physical design of 10 benchmarks whose gate count ranged anywhere from ~100 to ~60,000 standard cells. Each benchmark was synthesized at a number of different constraints to determine both the area-versus-delay tradeoff, as well as the leakage-power-versus-delay tradeoff.

For example, Figures 5.6 and 5.7 illustrate these tradeoffs for a Viterbi Decoding circuit (with ~25,000 gates). There are a few interesting points to notice from these plots.

First of all, the STEEL version has a better area/delay tradeoff characteristic. Hence, for the same critical path delay, the STEEL implementation will consume less area. This improvement occurs because the STEEL cells are identical in area to the traditional cells, but have reduced propagation delays (due to the stress-enhancement achieved through active-area overlap). Consequently, the physical design tools do not have to size a given STEEL path as aggressively as its corresponding traditional path implementation, leading to reduced area consumption.

Alternatively, if you analyze the circuits at the same value of area (iso-area), STEEL typically reduces delay by 11% (again, due to the stress-enhancement achieved without increasing area). Notice that even at the minimum delay point on the traditional curve, the STEEL library still provides ~9% improvement. Furthermore, if you examine the leakage tradeoff in Figure 5.7, leakage power in the Viterbi decoder increases rapidly on the left side of the plot (toward smaller values of delay). This is due to the fact that to meet these tight timing constraints, the synthesis tool must size up the majority of the gates in the design, which increases leakage dramatically. Since stress-enhanced gates are designed to

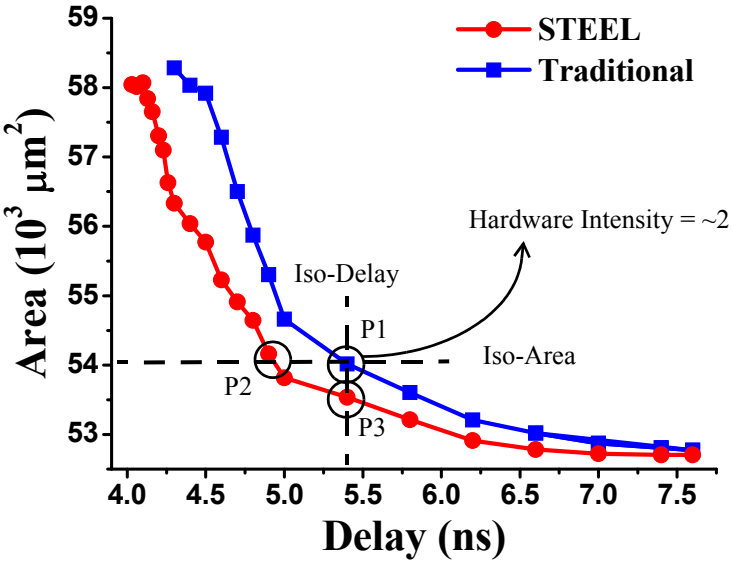


Figure 5.6. Viterbi Decoder Area vs. Delay (Single-V_{th}).

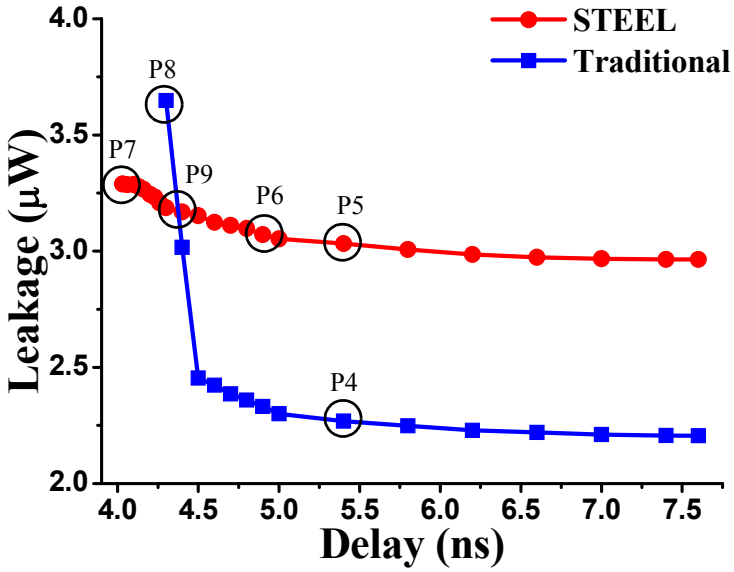


Figure 5.7. Viterbi Decoder Leakage Power vs. Delay (Single-V_{th}).

primarily give improvements in I_{on} (and therefore, delay), this region of the curve is where the STEEL library prefers to operate.

The full set of benchmark results compared to the single-RVT library is included in the seven left most columns of Table 5.1. This table was constructed using the following procedure. For each benchmark, we analyzed the area/delay tradeoff curve for the traditional 65nm implementation to determine the delay where hardware intensity was ~2. Hardware intensity was originally proposed in [73] as a power versus delay metric. In this work we use a modified version of hardware intensity that compares area and delay. Thus, for the remainder of the chapter, hardware intensity is defined as the percentage change in area over the percentage change in delay. Next, the corresponding values of area and delay (whose hardware intensity is ~2) were used to determine the iso-area and iso-delay comparisons made against the STEEL implementation. For example, in the Viterbi decoder benchmark, the point on the area/delay curve (for the traditional implementation) where the hardware intensity was equal to 2 is labeled point "P1" in Figure 5.6. The

Table 5.1. Design Improvement Obtained using STEEL.
(Compared against Single-V_{th} and Dual-V_{th} Implementations)

Circuit	Gate Count	% Delay Improvement (Iso-area)	% Area Improvement (Iso-delay)	Leakage Increase (Iso-delay)	Leakage Increase (Iso-area)	% Delay Improvement Beyond Min. Critical Path	$\frac{\text{Dual-Vth Leakage}}{\text{STEEL Leakage}}$ †
c432	143	18.6%	2.4%	1.41	1.46	12.5%	2.95
c1908	265	6.00%	6.7%	1.11	1.22	9.4%	4.88
c880	291	16.5%	2.6%	1.34	1.39	8.1%	2.37
c2670	489	9.2%	1.1%	1.35	1.34	4.4%	0.85
c3540	921	9.0%	2.1%	1.33	1.36	9.0%	2.08
c7552	1264	11.1%	0.9%	1.27	1.28	12.5%	2.97
c5315	1275	15.5%	1.5%	1.33	1.34	13.3%	2.78
c6288	1703	7.1%	0.4%	1.27	1.28	8.2%	3.52
Viterbi Dec.	25287	8.0%	1.1%	1.33	1.35	6.3%	2.06
Ethernet	66310	8.6%	0.1%	1.50	1.50	7.5%	0.79
AVERAGE		**11.0%**	**1.9%**	**1.32**	**1.35**	**9.1%**	**2.53**

† The dual-V_{th} leakage increase over STEEL is calculated at iso-delay for the minimum critical path delay of the STEEL design.

corresponding delay improvement that we achieve using STEEL is given in Column 3 of Table 5.1. For the Viterbi decoder, this value is calculated by comparing the delays at "P1" and "P2" (in Figure 5.6). Similarly, area improvement – Column 4 in Table 5.1 – is calculated by comparing the areas at "P1" and "P3". Next, Columns 5 and 6 include the leakage power increase incurred by the STEEL implementation. These values are calculated for the Viterbi circuit by comparing the leakage values at "P4" and "P5" (from Figure 5.7) for the iso-delay case, and comparing "P4" with "P6" for the iso-area column. Finally, the decrease in the minimum critical path delay is noted in Column 7. This value for the Viterbi decoder is determined by comparing the delay at points "P7" and "P8" in Figure 5.7. The remainder of Table 5.1 is discussed in Section 5.3.3.

Generally, we discovered that for iso-area, the STEEL implementation achieves average delay improvements of 11% while leakage only increases by 35% on average.

Additionally, we found that the STEEL-based benchmarks successfully synthesized at a minimum delay value that was, on average, 9.1% less than the traditional minimum delay.

5.3.3 STEEL versus Dual-V_{th} Results

In addition to a significantly improved area-delay tradeoff for STEEL versus a single-V_{th} standard library, we now demonstrate that *STEEL provides superior performance with a single-V_{th} over a traditional dual-V_{th} library* for the majority of operating points where dual-V_{th} would be of interest. This arises due to the improved I_{on} vs. I_{off} tradeoff using stress enhancement compared to using low-V_{th} devices (discussed in Section 4.3.2) and indicates that STEEL simultaneously offers a better power/performance envelope and lower manufacturing costs over dual-V_{th}. Figure 5.8, for example, illustrates the leakage/delay curve for the dual-V_{th} implementation of the Viterbi decoder (notice its similarity to Figure 5.7). The slower part of the curve (delay > 4.26ns) is actually identical to Figure 5.7, due to the fact that only RVT cells are used in the design until the delay

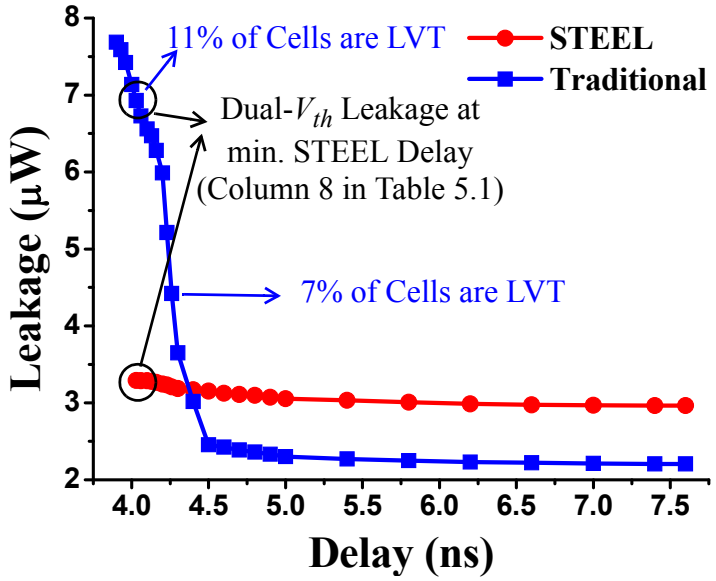


Figure 5.8. Viterbi Decoder P_{leak} vs. Delay Plot comparing Dual-V_{th} and STEEL.

constraint becomes less than or equal to 4.26ns. In the region of interest for STEEL, we found that the leakage crossover point (where dual-V_{th} leakage becomes greater than STEEL) typically occurred between the most tightly constrained RVT design (with zero LVT cells) and the dual-V_{th} implementation that used the minimum number of LVT cells needed to satisfy timing. Since the LVT cells in our industrial library typically increased leakage by ~20X, the minimum leakage for the dual-V_{th} case occurred at the timing constraint that used the minimum number of LVT cells. Even at this minimum leakage point for dual-V_{th} (where the number of LVT cells is only a small percentage of the total number of cells, <5%), the substantial leakage increase per low-V_{th} cell caused this minimum-leakage, dual-V_{th} implementation to almost match the leakage increase incurred by STEEL. Over all of the benchmarks, we found that even at the minimum dual-V_{th} leakage, dual-V_{th} only showed a 2.9% average savings in leakage over STEEL. Furthermore, by the time the STEEL implementations reached their minimum delay, the dual-V_{th} leakage had increased to ~2.5X the average value of STEEL leakage (displayed in the last column of Table 5.1). An example point for the Viterbi decoder circuit for this value is shown in Figure 5.8.

Since the STEEL implementations can typically provide up to ~10% delay improvements over single-V_{th} designs while consuming only a fraction of the leakage power of dual-V_{th}, STEEL can provide more optimal designs in two ways. First, for designs that only need moderate delay improvements – less than 10% – STEEL can be used to achieve these improvements. By utilizing the STEEL standard cells, the designer would not only reduce leakage (as compared to the dual-V_{th} implementation), but would

also dramatically reduce manufacturing costs, since the second threshold voltage mask would not be needed. Alternatively, STEEL could also be used in conjunction with the dual-V_{th} approach to achieve more optimal designs (in terms of area and power). Since typical dual-V_{th} processes only provide coarse-grain threshold voltage values, some standard cells in a path might be sub-optimally assigned if they do not need the full performance enhancement provided by moving to a lower V_{th} value. For these cells, the STEEL versions would be more appropriate, since they can obtain more fine-grained performance improvements and will fill some of the performance space between V_{th} values. Additionally, by designing LVT STEEL cells, delay improvement can be extended beyond the performance of dual-V_{th}.

5.3.4 Intelligent STEEL-Cell Assignment

One interesting discrepancy that we found during this work was the fact that in our largest circuit, an ethernet controller, the STEEL design did not outperform the dual-V_{th} implementation. In fact, out of the 10 benchmarks, the ethernet circuit was the only case where we did not obtain improvements in leakage versus dual-V_{th}. To understand this phenomenon, we analyzed the structure of the ethernet controller and made some interesting observations:

- Even though the ethernet controller used a large number of standard cells, its paths were not balanced and the number of critical paths only represented a small fraction of the total number of paths.
- Out of ~66,000 standard cells, the dual-V_{th} design only used 285 LVT cells (<1% of the total) to meet the minimum timing constraint achieved using STEEL.

With this knowledge, it was clear why the STEEL implementation did not improve upon the dual-V_{th} case. Since we had not previously employed any delay/leakage optimization in our approach, the ~1.3X STEEL average leakage increase per standard cell occurred in each of the ~66,000 standard cells, whereas the ~20X leakage increase per LVT cell only occurred in <1% of the total cells. Therefore, while the STEEL designs outperformed dual-V_{th} in the majority of our experiments, it was clear that exploring intelligent assignment schemes would be beneficial to our work, both to improve the STEEL leakage performance in unbalanced designs (as compared to dual-V_{th}), as well as achieve leakage values closer to the RVT-based designs.

So far, we have reported the STEEL results for the case where we use our stress-enhanced library uniformly across a given design (i.e, every gate in the circuit is assigned to its stress-enhanced version). However, not all of the gates in a circuit need performance enhancement to meet timing for a given delay constraint. These non-critical gates only add to the leakage overhead, and as a result we observed that the STEEL designs had larger leakage than their single-V_{th} counterpart, even at larger values of delay (more relaxed delay constraints). Thus, there is ample scope for intelligent assignment of stress-enhanced cells, where the traditional RVT library is used in conjunction with STEEL, and the STEEL cells are only assigned to timing critical gates. An intelligent cell assignment scheme will substantially reduce the leakage overhead but maintain similar improvements in delay. The benefits of this technique derive from the fact that only a fraction of total number of gates in a circuit are timing critical. Replacing only the critical gates with the leakier, higher-performance versions will result in significantly lower leakage increases, as compared to the case where all of the gates are replaced.

As a further investigation into the scope of intelligent assignment, we perform a simple experiment where we replace only the top ~10%, timing critical gates in a circuit with their stress-enhanced versions. We perform this experiment at the same hardware intensity point (discussed previously) on the area-versus-delay curve for the traditional RVT library, and compare the delay improvement and leakage overhead numbers to the case where stress enhancement was used in every cell (Column 3 and Column 6 of Table 1, respectively). Figure 5.9 shows the percentage improvement that we observe using intelligent assignment, as compared to the uniform-replacement ("Original" STEEL) scheme. Ideally, we would prefer to obtain all of the delay improvement achieved in the previous section (i.e., achieve 100% of the typical 11% delay improvement over RVT), while reducing the percentage leakage increase to 0% (i.e., matching the RVT leakage). As shown in the figure, we can get >80% of the "Original" delay improvement through selective replacement, while incurring a much smaller increase in leakage. The selective scheme typically reduces the uniform STEEL leakage increase by ~90%. From Figure 5.9, observe that the leakage number for the ethernet benchmark is exceptionally small

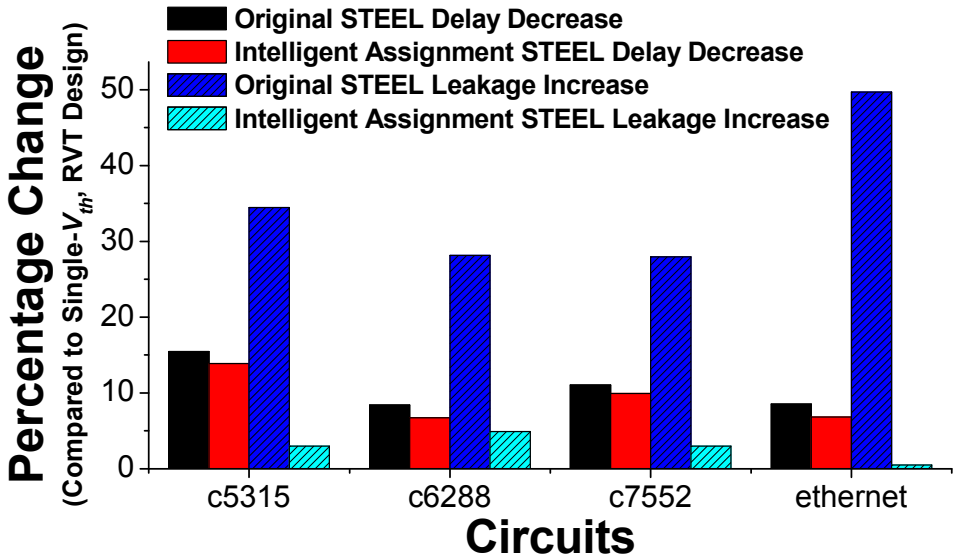


Figure 5.9. Impact of Intelligent STEEL Assignment on Delay and P_{leak}.

because, despite its large size (~66,000 gates), the number of timing critical gates is very small (as mentioned previously). Thus, to achieve 80% of the "Original" improvement, only 625 gates need to be replaced with their stress-enhanced version (less than 1% of the total gates), which results in substantial leakage savings that is comparable with dual-V_{th}.

Intelligent replacement schemes like this approach allow STEEL to maintain its advantage over dual-V_{th}, even for designs that are extremely unbalanced (such as the ethernet benchmark). Additionally, this approach can be used to improve leakage power consumption within any STEEL design (especially for relaxed delay constraints). This means that the leakage for the STEEL technique will approach that of the traditional RVT library, especially at delay constraints located to the right of the leakage crossing point (e.g., all of the STEEL leakage values to the right of point "P9" in Figure 5.7 will be much closer to RVT).

5.4 Summary

In this chapter, we proposed STEEL, a new standard cell library design technique for modern stress-enhanced semiconductor processes. STEEL fully exploits the layout dependencies of stress. By designing the STEEL standard cells to share the V_{DD} and V_{SS} source/drain connections across cell boundaries, one can achieve drive current improvements of up to 20%. While implementing the proposed standard cell approach in a number of benchmark circuits, we demonstrated average delay reductions of 11% with only a 35% average increase in leakage, compared to single-V_{th} implementations. Additionally, STEEL-based circuits typically achieved a ~2.5X reduction in leakage when

compared to dual-V_{th} designs. This implies that for designs requiring an 11% delay improvement (or less) beyond a single-V_{th} implementation, STEEL can provide this improvement for a smaller leakage penalty as well as much lower manufacturing costs compared to dual-V_{th}. Orthogonally, STEEL can also be used in conjunction with dual-V_{th} (similar to the work in Chapter 4) to provide more optimal designs (in terms of both leakage and delay).

CHAPTER 6

COMBINING STRESS ENHANCEMENT WITH GATE LENGTH BIASING

The previous two chapters presented the idea of improving mechanical-stress-induced mobility enhancement in today's transistors by modifying common circuit layout properties that influence stress. Mobility enhancement has emerged as one of the most prevalent manufacturing changes in recent semiconductor history because of its ability to enable continued process scaling. However, it is the optimization potential of mobility enhancement that has attracted a number of researchers, especially since designers are becoming increasingly wary of varying threshold voltage (V_{th}) in their circuits. Using multiple values of V_{th} is not as straightforward or beneficial in today's technologies, due to the amount of inherent uncertainty in threshold voltage and the extra mask cost incurred by including multiple V_{th} values in a design. Precisely controlling the value of V_{th} in modern-day processes is extremely difficult since the underlying sources (e.g., random dopant fluctuation, line-edge roughness, and work-function variation [75]) are truly random sources of variation inherent to current CMOS manufacturing.

Threshold voltage optimization is, at its core, merely a tradeoff between steady-state power consumption and performance. Generally, lowering a transistor's V_{th} means that

that the transistor will switch states faster, but will consume exponentially higher amounts of power in steady-state. Since the magnitude of steady-state power consumption (also called leakage power consumption) in state-of-the-art circuits is approaching the same order of magnitude as dynamic power consumption (shown previously in Figure 1.3 and repeated here as Figure 6.1, for convenience), V_{th} optimization has largely been used as a leakage savings technique; choosing a slower device with a higher V_{th} saves exponentially in leakage. However, the difficulties encountered by modern-day process engineers in controlling threshold voltage have lead circuit designers to explore other leakage savings techniques. Two such techniques that, until this work, have previously been explored independently are gate length biasing and mechanical stress optimization.

Mechanical stress optimization is a technique that involves leveraging the mechanical-stress-dependent layout properties in a circuit to vary mobility and ultimately increase/decrease performance while increasing/decreasing leakage. By manipulating properties like active area, gate-to-contact spacing, and active edge placement (relative to the lateral STI), layout designers can achieve maximum performance gains of 10 – 20%.

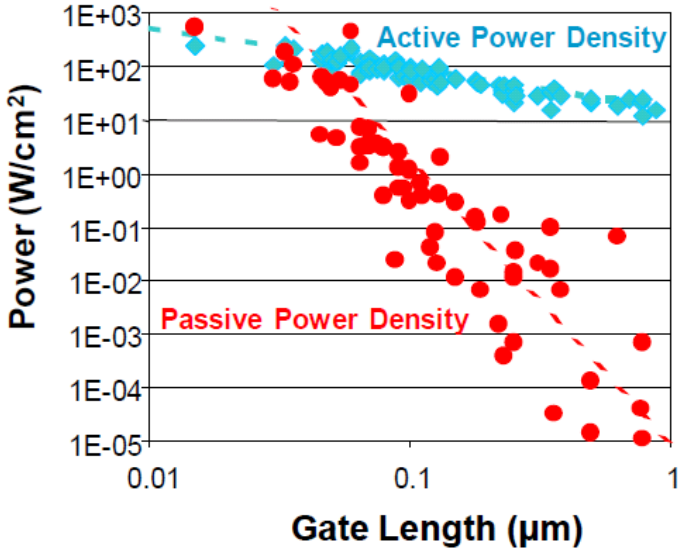


Figure 6.1. Dynamic and Static Power Density vs. Technology [19].

This performance gain becomes especially appealing after discovering that the resulting leakage penalty is substantially less (typically ~2X) than the penalty incurred by an equivalent dual-V_{th} implementation. For a full background on mechanical stress optimization, the reader is referred to Chapters 4 and 5, where mechanical stress was covered in great detail.

On the contrary, gate length biasing is a more established technique that involves increasing (decreasing) a transistor's gate length to simultaneously decrease (increase) its performance and leakage power consumption. The idea was first proposed in [76], where the authors used large increases in gate length (up to ~50nm) to reduce leakage. Then, in [77], the approach from [76] was amended to only use small (8nm) gate length biases. In this approach, 8nm biases were chosen to maximize the leakage savings while still allowing the gate-length biased cells to be layout-swappable with their higher performance counterparts.

In order to meet equivalent delay targets achieved by dual-V_{th} (DVT) schemes, previous gate-length bias (GLB) works (such as [77]) primarily used GLB in conjunction with DVT optimization, since negative gate-length biases (i.e., smaller than nominal gate lengths) are typically not allowed by manufacturers (since smaller gate lengths increase leakage and are more susceptible to short channel effects and variability). However, in this work we noticed that layout-dependent mechanical stress enhancement could be used along with GLB to provide a competitive optimization alternative to DVT, in terms of performance, while reducing leakage power consumption (since stress-enhancement has a better delay/leakage tradeoff than DVT). Additionally, using stress-enhancement and GLB

instead of DVT reduces mask costs (due to the elimination of the additional V_{th} mask) and overall variability [77].

Therefore, this chapter illustrates the benefit of using layout-dependent stress enhancement and GLB versus DVT. Since Chapters 4 and 5 went into great detail discussing the benefits of stress-enhancement versus DVT, this chapter only outlines layout-dependent stress enhancement and GLB (in Section 6.1). Section 6.2 explains how stress-enhancement and GLB can be combined in standard cell library design and then describes the standard cell implementation for our stress plus GLB library (referred to as STLB from this point forward). The optimization algorithm written to utilize the STLB library is discussed in Section 6.3 and Section 6.4 illustrates the results obtained by using STLB optimization on eight different benchmark circuits (six ISCAS'85 circuits and two larger Viterbi decoding circuits). Finally, Section 6.5 concludes the chapter with a brief summary.

6.1 Stress Enhancement and Gate-Length Biasing

From equations (4–1) and (4–2) in Chapter 4 (and from basic semiconductor classes), we know that a device's saturation ($I_{D,sat}$) and subthreshold ($I_{D,sub}$) drain currents are both functions of a number of parameters (for convenience, the two equations are also copied below), including carrier mobility (μ_0) and gate length (L_{eff}, which we refer to as L for the remainder of the chapter).

Table 6.1. Methods for Increasing PMOS and NMOS Mobility in Standard Cells.

	Active Area	Gate-to-Contact Spacing	Active Edge to Lateral STI Spacing
PMOS	Increase	Increase	Decrease
NMOS	Increase	Increase	Increase

$$I_{D,sat} = \frac{\mu_0}{[1 + U_0(V_{GS} - V_T)]} \cdot \frac{C_{ox}}{2aV} \cdot \frac{W}{L_{eff}} \cdot (V_{GS} - V_T)^2$$
$$V = \frac{1 + v_c + \sqrt{1 + 2v_c}}{2} \qquad v_c = U_1((V_{GS} - V_T)/a) \tag{6–1}$$

$$I_{D,sub} = A \cdot e^{\frac{1}{nv_T} \cdot (V_G - V_S - V_{th0} - \gamma' V_S + \eta V_{DS})} \cdot (1 - e^{(-V_{DS})/v_T})$$
$$A = \mu_0 C_{ox} \frac{W}{L_{eff}} v_T^2 e^{1.8} e^{-\frac{\Delta V_{th}}{\eta v_T}} \tag{6–2}$$

Specifically, a device's drain current (in both regimes of operation) is directly dependent on μ_0 and inversely dependent on *L*. In Chapters 4 and 5, we learned that carrier mobility in modern-day devices can be increased or decreased by changing certain transistor layout parameters. For example, modifying properties like active area, contact placement, and active edge placement (with respect to the lateral STI) alters the mechanical stress induced in a transistor's channel which, in turn, affects μ_0. Prior to the 90nm technology node, the only device properties that a designer could use to significantly affect performance were *L* and *W*. However, with the addition of mechanical stress, engineers now have a third parameter to manipulate: carrier mobility. After identifying the dominant layout dependencies of stress in a given technology, designers can utilize those dependencies to modify mobility. For example, in the 65nm industrial technology presented in the previous chapters, modifying one or more of the layout properties in Table 6.1 (as prescribed in the

table) will increase channel mobility. This allows transistors with minimum gate lengths to increase performance (and leakage) without increasing the transistor's width (W in equations 6–1 and 6–2).

Gate-length biasing, on the other hand, directly manipulates the gate length, L, of a transistor. In traditional digital circuit design, gate length is typically minimized for a number of reasons: smaller L means faster switching, less gate capacitance, and less dynamic power consumption. Therefore, in a given technology, digital designers usually desire the smallest possible gate length and process engineers strive to provide the smallest gate possible while simultaneously optimizing performance, leakage current, and printability (which affects yield). Tuning a process's minimum gate length, however, is becoming increasingly difficult because as transistors continue to shrink from generation to generation, short channel effects (SCE) are having a larger impact on performance and leakage. Thus, gate-length biasing has emerged as a popular technique (embraced by a number of companies [77]) to help combat SCE and improve variability. GLB is a viable technique because every transistor in a circuit does not require the speed provided by the high performance, minimum L transistors (a trait DVT and stress-optimization also rely on). By slightly increasing (biasing) the L of non-critical transistors, circuit designers can save leakage power and improve variability while minimally increasing area and dynamic power consumption. In [77], the L biases proposed were <10% of $L_{nominal}$ because the leakage savings saturated around 10%. Similarly, in the 65nm technology used in this work, we discovered that the leakage savings also saturated around 10%, as shown in Figure 6.2. Note that in Figure 6.2, I_{ON} and L are a normalized percentage about the

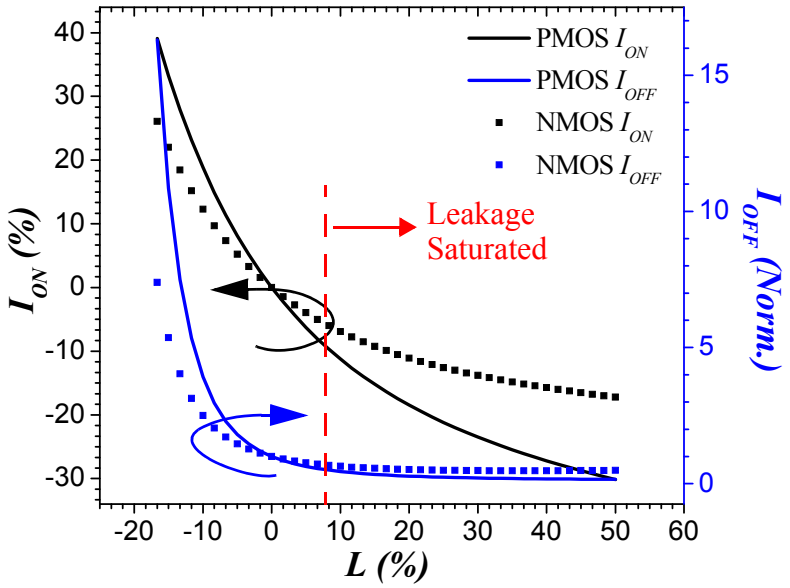


Figure 6.2. Normalized I_{on} and I_{off} vs. L for PMOS and NMOS devices.

nominal value (I_{ON} at $L = L_{nominal}$), while I_{OFF} is a normalized, scaled value of its nominal (I_{OFF} at $L = L_{nominal}$). After understanding the gate length bias impact in our 65nm technology, we were able to design and optimize a standard cell library that simultaneously contained stress-enhancement and gate length biasing.

6.2 STLB Standard Cell Library Implementation

This section explains how we combined stress-enhancement and gate-length biasing, and then presents our STLB standard cell library implementation.

6.2.1 Combining Stress-Enhancement and Gate-Length Biasing

The overall goal of this work was to study and compare an STLB library to its DVT counterpart. Since stress sources (such as embedded SiGe, dual-nitride liners, and the stress memorization technique) are typically used in high performance processes,[1] we decided that the most appropriate comparison would be a high performance one:

- Dual-V_{th}: low-V_{th} (LVT) cells and regular-V_{th} (RVT) cells

versus

- STLB: high-stress (HST) cells with low-stress, +5nm length biased (GLB) cells.

Both optimization schemes attempted to maximize performance while minimizing leakage power consumption. In the STLB library, we used layout properties (such as the column headers in Table 6.1) to increase mobility (through stress) and, consequently, performance. With stress-based mobility enhancement, we were able to increase the performance of our regular-V_{th} (RVT) standard cells anywhere from 5% to 15%. In order to achieve adequate performance increases in these cells, we increased the area of the RVT cells by 24% on average (each standard cell width was increased by one metal track). The low leakage cells, on the other hand, were low-stress, +5nm biased RVT cells. By manipulating the stress-dependent layout properties of the RVT cells conversely to Table 6.1 while adding a +5nm gate length bias (which was close to the "knee" of the curve in Figure 6.2, where leakage savings saturated), we were able to maximize the leakage savings of our GLB cells. The next section lists the specific characteristics of the STLB library and compares its performance to the DVT implementation.

6.2.2 The STLB 65nm Library

To create our 65nm STLB library, we used industrial 65nm standard cells as our baseline. Then, for 10 basic standard cells in the library, we created the stress-enhanced (high-stress, or HST) layouts by following the guidelines presented in Table 6.1. The GLB

[1] Stress-based mobility enhancement is used more liberally in high performance processes because the resulting increase in leakage power is typically unattractive for most low power processes.

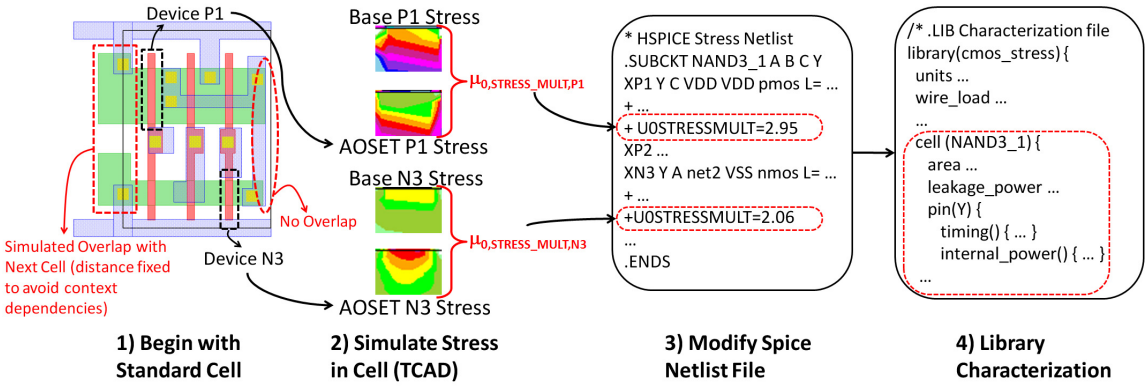


Figure 6.3. STLB characterization flow.

layouts, on the other hand, were created by reversing the guidelines from Table 6.1 to ensure minimal stress-enhancement and then increasing the gate lengths by 5nm. For the HST cells, each layout was simulated in Synopsys's *Tsuprem4* (which simulated the device fabrication steps and calculated the stress) and *Davinci* 3D (which simulated the electrical operation and computed mobility) TCAD tools to evaluate the mobility enhancement achieved in the new layouts. Similar to Section 5.2, the mobility enhancement factors calculated for each transistor were written into a modified BSIM4 Hspice model (the details of which can be found in Section 5.2.2). Finally, this modified Hspice model was used by Cadence's *Encounter Library Characterizer* (ELC), which generated the LIBERTY format file that contained performance information such as propagation delays, power consumption, etc., for each cell. Overall, the HST library characterization flow was similar to Figure 5.3, which has been reproduced here as Figure 6.3, for convenience. Cadence's ELC tool was also used to characterize the GLB library.

When we compared the LIBERTY files generated by the ELC, the benefit of using STLB over DVT was apparent. While the DVT library provided a larger spread in performance (~27% when averaging rise and fall $\frac{LVT}{RVT}$ performance), it also incurred a

much larger spread in leakage power consumption (~13X). The STLB library, on the other hand, provided a 12% performance difference between the HST cells and the GLB cells for only a 2.5X difference in leakage. Furthermore, we directly compared the high performance cells (LVT and HST) separately from the low performance cells (RVT and GLB), and found that the LVT cells had ~22% better delay than the HST cells, on average, but they consumed 7.6X more leakage power. Similarly, the RVT cells had ~6% better delay than their GLB counterparts, while consuming 1.5X more leakage, on average.

Table 6.2 illustrates similar types of comparisons across three difference cells (one inverter, one 2-input NAND, and one 2-input NOR) from our two libraries, and also includes an average comparison across all 10 cells. Four ratios are shown for each cell. The first two of these ratios show a direct comparison between the two different libraries: $\frac{LVT}{HST}$ and $\frac{RVT}{GLB}$. These rows allowed us to examine the performance difference between each class of cell (high performance or low performance). The remaining two ratios illustrate the performance difference in one library: the $\frac{LVT}{RVT}$ row shows the spread in performance for the DVT library, while the $\frac{HST}{GLB}$ row shows the spread in performance for the STLB library. Table 6.2 contains the performance ratios for six different parameters:

- Rise delay is the 50%-50% propagation delay for a rising output
- Fall delay is the 50%-50% propagation delay for a falling output

Table 6.2. DVT vs. STLB Library Comparison.
(Note: DVT vs. STLB cell comparison is shaded; DVT and STLB spread are unshaded.)

	Ratio	Rise Delay	Fall Delay	Leakage Power	Internal Power	Dynamic Power	Area
INVX1	LVT/HST	-19.2%	-18.0%	5.9X	1.1X	3.0%	-25.0%
	RVT/GLB	-5.4%	-6.1%	1.5X	1.5X	-2.2%	0.0%
	LVT/RVT	-24.6%	-22.8%	10.4X	2.0X	5.0%	0.0%
	HST/GLB	-11.5%	-11.5%	2.6X	4.5X	-0.3%	33.3%
NAND2X4	LVT/HST	-18.0%	-28.9%	6.9X	1.2X	4.2%	-12.5%
	RVT/GLB	-6.1%	-5.4%	1.6X	1.1X	-3.1%	0.0%
	LVT/RVT	-22.7%	-32.2%	16.7X	2.3X	5.6%	0.0%
	HST/GLB	-11.3%	-10.1%	3.8X	1.6X	-1.9%	14.3%
NOR2X4	LVT/HST	-19.0%	-30.0%	8.6X	1.9X	5.1%	-12.5%
	RVT/GLB	-5.7%	-6.8%	1.4X	1.2X	-2.5%	0.0%
	LVT/RVT	-26.0%	-31.8%	14.7X	2.1X	4.5%	0.0%
	HST/GLB	-13.8%	-10.1%	2.4X	1.3X	-3.1%	14.3%
AVERAGE (across all 10 cells)	**LVT/HST**	**-18.1%**	**-25.4%**	**7.6X**	**1.5X**	**3.4%**	**-18.8%**
	RVT/GLB	**-5.7%**	**-6.1%**	**1.5X**	**1.2X**	**-2.5%**	**0.0%**
	LVT/RVT	**-24.2%**	**-29.1%**	**12.8X**	**2.3X**	**4.2%**	**0.0%**
	HST/GLB	**-12.6%**	**-11.4%**	**2.5X**	**1.8X**	**-1.7%**	**23.5%**

- Leakage power is the average steady-state power consumed by the cell (across all input states)
- Internal power is the average power consumed during a transition that is internal to the cell (e.g., short circuit current, internal switching capacitance, etc.)
- Dynamic power is derived from a cell's increase in input pin capacitance, since pin capacitance affects the upstream dynamic power of the preceding gates
- Area is the area consumed by the standard cell.

It is interesting to note that DVT also incurred a slight increase in the internal power consumption for both the high performance (LVT) and low performance (RVT) cells (about 50% and 20%, respectively). This increase can most likely be attributed to larger short circuit current in the LVT and RVT cells, compared to their HST and GLB

counterparts. The difference in dynamic power, however, is much smaller: LVT consumed 3.4% more dynamic power than HST while GLB consumed 2.5% more dynamic power than RVT. These numbers were derived directly from the input pin capacitance increase. Since the LVT and GLB cells had larger pin capacitance (due to the lower V_{th} and larger L, respectively), those cells created larger dynamic power consumption for their fan-in. Lastly, the only cells that increased in area were the HST cells, which were ~24% larger (on average) than the LVT, RVT, and GLB cells, because of the additional metal track space added to the width of each cell, discussed previously.

6.3 Dual Performance Optimizer for DVT and STLB Libraries

In order to obtain circuit-level comparisons between DVT and STLB libraries, this work required a custom, dual-performance optimization algorithm, similar to the optimization methodology presented in Section 4.7. However, as stated earlier in the chapter, since stress enhancement is used in high performance processes, the proposed algorithm in this chapter differs from Algorithm 4–1 in that it strives to achieve the best delay possible and then minimize leakage power consumption once that delay is met. In this algorithm, there are two types of cells available: high-performance cells and low-leakage cells. The algorithm begins by setting each gate to its high performance version (e.g., LVT or HST, depending on which library is being used). Then a typical STA and sizing algorithm is used to meet the defined cycle time for the circuit. Next, the dual-performance optimizer is called. Each iteration, it identifies one gate that, when upsized, provides the largest improvement in delay (thereby creating additional slack in the

circuit). After the particular gate is upsized, the optimizer evaluates a merit function – shown below in (6–3) – for every gate in the circuit.

$$Merit(G) = \Delta I_{off}(G) \cdot Slack_{\alpha} \quad \text{where } \alpha \text{ is the path that contains } G \tag{6–3}$$

The optimizer then replaces the highest merit gates with their low-leakage versions and calculates the resulting total power improvement. If the circuit power improves, the move is accepted and the optimizer attempts to find another gate to upsize, which will allow more gates to be replaced by their low-leakage versions. The optimizer halts once all potential low-leakage replacements have been evaluated and upsizing no longer creates slack in the circuit. The pseudo-code for a given critical delay target (T_T) is shown in Algorithm 6–1.

Algorithm 6–1 DELAY_LEAKAGE_OPT(T_T) //T_T = critical delay target

```
1:  Set all cells in netlist to High Performance version (e.g., LVT or HST)
2:  // Initialize circuit and run sizing algorithm to meet critical delay target
3:  T_C = SIZE_CIRCUIT_TO_MEET_DELAY(T_T) // T_C = current CP delay
4:  if ( T_C == T_T ) //Then timing constraint is met, perform leakage opt.
5:      Evaluate current power consumption // P_TOT = total power
6:      P_NEW = P_TOT
7:      while ( P_NEW <= P_TOT )
8:          Order gates by potential delay improvement after upsizing
9:          Upsize first gate // Has largest delay improvement
10:         Evaluate MERIT(G) for all gates, G // According to (6–3)
11:         Move highest merit gates to low-leakage versions (e.g., RVT or GLB)
12:         Evaluate new T_C and P_NEW
13:         if ( (T_C <= T_T) AND (P_NEW < P_TOT) ) // Then accept move
14:             P_TOT = P_NEW
15:         else
16:             // Undo low-leakage moves and last gate upsize
17:             Restore previous state
18:         end if
19:     end while
20: end if
```

6.4 Experimental Results

As stated previously, the ultimate goal of the work in this chapter was to obtain a circuit-level comparison of a DVT library versus an STLB library. To achieve this, we used the custom, dual-performance algorithm described earlier in Section 6.3 to optimize a number of ISCAS'85 benchmarks and two larger Viterbi decoding circuits. This section presents the comparison results for the eight largest circuits (six ISCAS'85 benchmarks and the two Viterbi decoders). For each benchmark and library (DVT or STLB), we optimized the circuit across a number of critical path delay targets. Then we compared the delay, area, leakage, and dynamic power performance of each library. The comparison points discussed in the remainder of this section were once again chosen based on a modified version of hardware intensity [73].

From Section 4.8.2, we know that a hardware intensity (η) of x means that a 1% decrease in delay leads to an x% increase in power, and the hardware intensity for the majority of blocks in a microprocessor design is between 2 and 3 [74]. Thus, the performance results presented in this section (and specifically in Table 6.3) were compared at the minimum delay point where the hardware intensity was between 2 and 3 (or as close as possible). To visualize the performance of each library, refer to Figures 6.4 – 6.6 for an example comparison. Figures 6.4 through 6.6 show critical path delay versus leakage power, area, and dynamic power for the Viterbi Decoder 1 benchmark.

As depicted in Figure 6.4, the leakage-based hardware intensity point for the Viterbi Decoder 1 benchmark was located at the normalized delay values of 1.4 and 1.5 for the

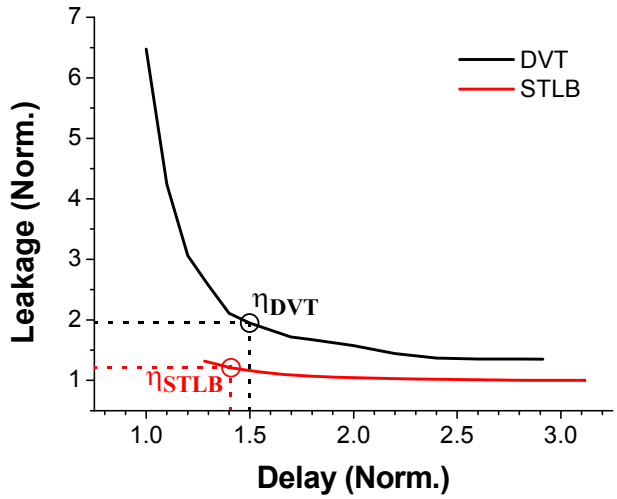


Figure 6.4. Normalized Leakage Power vs. Delay for Benchmark Viterbi Decoder 1.

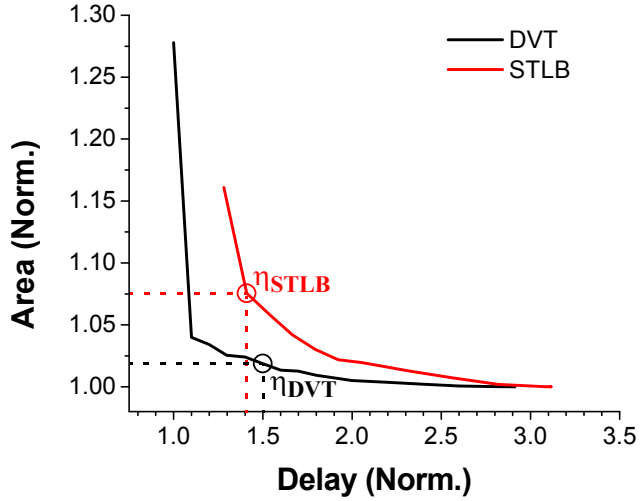


Figure 6.5. Normalized Area vs. Delay for Benchmark Viterbi Decoder 1.

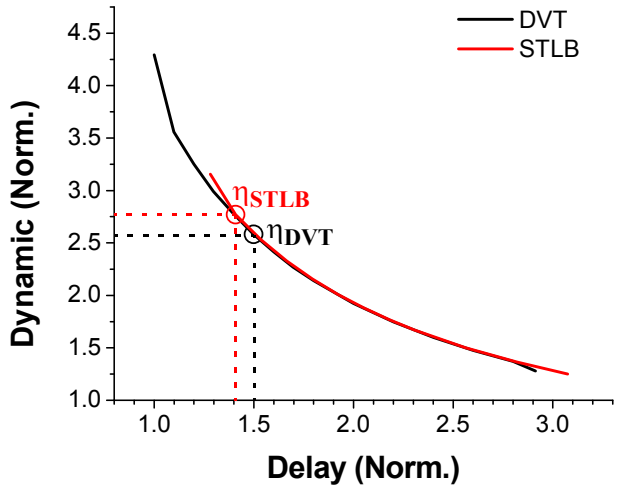


Figure 6.6. Normalized Dynamic Power vs. Delay for Benchmark Viterbi Decoder 1.

Table 6.3. STLB Performance Directly Compared to DVT.[a,b]

Benchmark	# Gates	Delay	Leakage Power	Dynamic Power	Area
c1908	499	0.98	3.23	0.98	0.8
c2670	885	0.84	3.49	1.19	0.91
c3540	1193	1.07	2.58	0.88	0.84
c5315	1730	0.98	2.23	1.01	0.92
c6288	2598	0.89	2.15	1.12	0.93
c7552	2060	1.05	1.99	0.93	0.79
Viterbi Dec. 1	12181	1.08	1.61	0.90	0.91
Viterbi Dec. 2	33972	0.92	5.73	1.05	0.79
AVERAGE		**0.98**	**2.88**	**1.01**	**0.86**

a. Comparison made at minimum delay point where hardware intensity was between 2 and 3.

b. Each column compares performance as a ratio: $\frac{DVT}{STLB}$.

STLB and DVT libraries (respectively). The STLB implementation, therefore, was almost 8% faster. Additionally, the STLB circuit consumed ~1.6X less leakage than its DVT counterpart. The STLB implementation provided these benefits while only increasing area and dynamic power by about 9% and 10% (compared to DVT). These increases were expected, due to the area overhead incurred by the HST cells and the pin capacitance increase of the GLB cells.

Table 6.3 shows the full comparison for all eight benchmarks. The circuits' gate count ranged from ~500 gates to ~34,000 gates. Overall, the STLB library delay averaged 2.4% slower than the DVT library (at the iso-hardware-intensity comparison point) but consumed ~2.9X less leakage power. As expected, the area of the STLB library was larger than the DVT library (by 14% on average), but the dynamic power was almost equal (the STLB library actually showed an average dynamic power improvement over DVT of 0.9%). The amount of area increase can be attributed to the higher area of the HST cells (referring back to Table 6.2, the HST cells were 23.5% larger in area, on average), as well

as the slower delay of the STLB cells. That is, the only way that the STLB library could match the DVT library's lower values of delay was to use more, higher area HST cells. However, since on-chip area in state-of-the-art processes is relatively inexpensive, designers are increasingly willing to tradeoff increased area for increased performance. Thus, the ~3X leakage reduction for a 14% area increase should be a worthwhile tradeoff.

6.5 Summary

In this chapter, we presented a joint optimization framework that used stress-enhancement with gate length biasing to maximize circuit performance (i.e., achieve small critical path delays) while minimizing leakage consumption. The resulting library, called STLB, was directly compared to DVT, both on the cell level and the circuit level. The STLB library provided a ~12% delay performance spread for a 2.5X leakage increase, whereas the DVT counterpart provided a ~27% delay spread for a 13X leakage increase (which, comparing the ratio of delay-spread to leakage-spread, was >2X smaller than STLB). On the circuit level, the STLB implementation typically came within ~2% of DVT, in terms of delay, and reduced leakage power consumption by ~2.9X (for an average area increase of ~14%). Therefore, providing that a <20% area increase is tolerable, combining stress-enhancement with gate length biasing can offer an excellent alternative to dual-V_{th}.

CHAPTER 7

CONCLUSION AND FUTURE WORK

As the semiconductor industry forges into the next decade, the integrated circuit design roadmap is as uncertain as ever. The manufacturing community is constantly working to push the scaling barrier lower, but physical fundamental limits decrease the performance gains traditionally achieved by CMOS scaling and threaten to halt scaling altogether around the 15nm technology node. Thus, in order to produce viable designs at next-generation nodes like 32nm, 22nm, and beyond, circuit designers and process engineers have to collaborate under the expanse of DFM. The work presented throughout this dissertation taught us that DFM is an essential semiconductor field that contains a plethora of broad, difficult problems. Since complex interactions now exist between how transistors are used, how they perform, and how they are manufactured, circuit design must also evolve to produce optimal results. That means that the models, tools, optimization schemes and processes all have to work together to understand all of the tradeoffs. The remainder of this section summarizes our contributions to these areas and concludes with a discussion of future work.

7.1 Conclusion – Summarizing Our Contributions

This dissertation primarily dealt with improving the awareness and accuracy of the underlying process models, Computer-aided Design (CAD) tools, and IC optimization schemes. The work presented in Chapters 2 and 3 focused on improving the spatial correlation models and variability models used with Statistical Static Timing Analysis. In the 0.13μm study in Chapter 2, we discovered that the Quad-tree correlation model generally outperformed four other prominent correlation models, especially as die size decreased. Another important observation made was that the simple correlation model that expressed CD as a function of two parameters – inter-die variation (which was perfectly correlated) and independent variation – came within 4% of the accuracy of the complex Quad-tree and PCA models, with significantly less overhead and run-time. The exploration in Chapter 2 emphasized the classic tradeoff between accuracy and efficiency and illustrated the need for judicious selection of correlation models within all timing analyses (both STA and SSTA).

Chapter 3, on the other hand, investigated the underlying process models essential to SSTA. Specifically, Chapter 3 proposed a new statistical model for gate length (or CD). During this work, we discovered that the current method for modeling CD within SSTA was error-prone and could sometimes cause twice as much error as total variation. The magnitude of this error was derived from the fact that the existing CD models did not capture the complex, context dependent interactions that arose in state-of-the-art processes. However, after using PCA to decompose CD variability within a standard cell library, we discovered that adding one random variable to the CD model would significantly improve accuracy, while minimally impacting the characterization

complexity and run-time. After implementing the proposed CD model on a 90nm standard cell library, we found that our model – which could easily be incorporated within existing SSTA frameworks – reduced standard deviation error by ~3X.

After improving the modeling within IC timing analysis, we shifted our focus to improving the optimization tools and schemes that allow designers to intelligently improve digital circuits. In this document, the optimization tools and schemes explored primarily involved mechanical-stress-based enhancement. Chapters 4, 5, and 6 all presented different stress-based methodologies that either worked with current dual-V_{th} frameworks, or aimed to replace dual-V_{th}, altogether.

Our mechanical stress optimization study began in Chapter 4. In this chapter, we described the potential for mechanical-stress-based enhancement and discovered the improved delay/leakage tradeoff of mobility enhancement (compared to DVT). We also identified the set of layout properties in a 65nm technology that allowed us to influence mobility through layout. Using those properties to improve performance, we then created a 65nm stress-enhanced library that was used in conjunction with DVT. By including stress-enhanced cells within a traditional DVT library, we were able to reduce leakage power consumption by ~24% without increasing delay (and only increasing area by <0.5%). Alternatively, we used the same framework to reduce delay by ~5% without increasing leakage (for the same, <0.5%, area penalty).

Since DVT design is becoming increasingly problematic with each subsequent process node, Chapters 5 and 6 investigated ways to use mechanical-stress-based mobility enhancement to replace DVT optimization. In Chapter 5, we proposed a new library

design methodology, called STEEL, which shared the V_{DD} and V_{SS} (power and ground) source/drain connections across standard cell boundaries and, consequently, increased mobility and performance (due to the strong active area dependency of mechanical stress). By sharing the power and ground connections across standard cell boundaries, we discovered that we could improve drive current by up to ~20% without increasing area. Overall, this standard cell performance improvement lead to circuit delay reductions of 11% while only increasing leakage by 35% – a 2.5X reduction from equivalent DVT implementations. Thus, STEEL was our first optimization exploration that attempted to create efficient, high performance circuits without modifying V_{th}.

The final study that we performed for this dissertation work was also our final study on high performance, stress-aware circuit optimization. In Chapter 6, we sought a more highly optimized solution that focused on leakage savings in high performance designs. This solution manifested itself as STLB, a library that combined high performance, highly stressed devices with low-leakage, gate length biased devices. Using stress-enhancement and gate length biasing, we were able to create a library and optimization scheme that more closely resembled DVT. The final STLB library implementation provided a ~12% spread in delay performance for a 2.5X spread in leakage consumption. After optimizing a set of circuits with the STLB library, we discovered that the STLB circuits typically came within ~2% of the DVT delay while reducing leakage power consumption by ~2.9X (for an average area increase of ~14%). Therefore, we determined that combining stress-enhancement with gate length biasing could offer an excellent alternative to dual-V_{th}, provided that a <20% area increase was tolerable.

7.2 Future Work

As alluded to at the beginning of this chapter, there are numerous problems within IC design and DFM that need to be addressed in current and future process nodes. In this section, we propose future explorations related to the work presented throughout this dissertation.

7.2.1 CD Modeling at Advanced Process Nodes

In Chapter 1, it was mentioned that the semiconductor industry currently uses 193nm wavelength light to produce sub-100nm transistors. The light source currently used to produce this wavelength light has not changed since around the 130nm process node, and likely will not change until somewhere around the 15nm node. Instead, manufacturers are using processing techniques such as resolution enhancement (e.g., OPC), double patterning/exposure, and immersion lithography to produce the sub-wavelength features. Since the lithographic system itself is not changing, the exposure window (the region of the wafer that is illuminated at any particular time) is also staying relatively constant. This means that at every node, when device area scales by ~1/2, almost twice as many devices can exist within the same exposure window. All of the patterns that exist within the exposure window interact to create various diffraction patterns, and ultimately influence the size and quality of the printed geometries. In the CAD community, these interactions are typically referred to as "context dependencies" at the layout-level. Therefore, a particular gate's size is not only a function of its designed size, but also a function of the characteristics of its neighboring gates. "Regular" standard cell design (e.g., logic bricks, fixed-pitch polysilicon, etc.) has emerged to aid in reducing the context variability, but

perfect regularity is costly to achieve. Thus, there is a tremendous need for intelligent context models on the design-side of the DFM space. This means that studies of advanced process node context should be conducted so that the results can be analyzed, and intelligent models can be developed. Ultimately, this context dependency should be included within variability models so that it can be accounted for by CAD tools at design-time.

7.2.2 Library Characterization, Automation, and Optimization

The discussion of context dependency in the previous section is just one example of the layout-level dependencies that occur in modern-day technology nodes. Other examples (mentioned throughout this work) include the stress-based mobility dependence and well proximity dependence present in today's processes. While creating state-of-the-art layouts, designers must now be aware of device context, active area size, gate-to-contact spacing, and a number of other parameters that all affect transistor performance. This is a large number of interactions to manage in every standard cell design, and the magnitude of the interactions appears to be changing at every subsequent process node. In order to truly optimize the 100's of standard cells that appear in today's libraries, some level of automation is needed in order to achieve a certain amount of design efficiency. Therefore, creating layout automation tools that understand all of the process-dependent layout parameters and their interaction is an interesting and important area of research that is essential for current and future process nodes.

7.2.3 Further Exploration of Mechanical Stress

Aside from incorporating the knowledge of mechanical-stress-dependent layout parameters into state-of-art layout automation tools, further exploration of the benefits and limits of mechanical stress is also needed. The mechanical-stress-based work presented in the previous three chapters relied on Technology-CAD (TCAD) tools to simulate and characterize the amount of mobility-enhancement achieved after manipulating the mechanical stress. However, the correlation between TCAD simulation and fabricated device measurements is a topic that has not been well published. In order to evaluate the full potential of stress-based mobility enhancement, actual silicon-based studies are needed. These silicon test chips should explore and validate a number of areas:

- the layout dependence of stress and its correlation to TCAD (especially with respect to the dependence on active area)
- the influence that various process structures (e.g., sigma-shaped eSiGe) have on mechanical stress and its layout dependencies
- the variability of mechanical-stress-based enhancement and its overall impact on performance variability (especially delay and leakage variability).

These types of studies are necessary to strengthen the case for stress-based-enhancement and numerous research studies will have to be conducted to truly compare its merits to dual-V_{th} optimization.

RELATED PUBLICATIONS

- B. Cline, K. Chopra, D. Blaauw, and Y. Cao, "Analysis and Modeling of CD Variation for Statistical Static Timing," in *Proc. IEEE/ACM Int. Conf. on Computer-Aided Design*, pp. 60–66, Nov. 2006.

- B. Cline, K. Chopra, D. Blaauw, A. Torres, and S. Sundareswaran, "Transistor-Specific Delay Modeling for SSTA," in *Design, Automation, and Test in Europe*, pp. 592–597, Mar. 2008.

- V. Joshi, B. Cline, D. Sylvester, D. Blaauw, and K. Agarwal, "Stress Aware Layout Optimization," in *Proc. Int'l Symp. on Physical Design*, pp. 168–174, Apr. 2008.

- V. Joshi, B. Cline, D. Sylvester, D. Blaauw, and K. Agarwal, "Leakage Power Reduction Using Stress-Enhanced Layouts," in *Proc. 45th Annu. Conf. on Design Automation*, pp. 912–917, June 2008.

- V. Joshi, B. Cline, D. Sylvester, D. Blaauw, and K. Agarwal, "Mechanical Stress Aware Optimization for Leakage Power Reduction," *IEEE Trans. on CAD of Integrated Circuits and Systems*, to appear 2010.

- B. T. Cline, V. Joshi, D. Sylvester, and D. Blaauw, "STEEL: A Technique for Stress-Enhanced Standard Cell Library Design," in *Proc. IEEE/ACM Int. Conf. on Computer-Aided Design*, pp. 691–697, Nov. 2008.

BIBLIOGRAPHY

[1] Intel Corporation. *4004 Single Chip 4-Bit P-Channel Microprocessor* [Online]. Available: http://download.intel.com/museum/archives/pdf/4004_datasheet.pdf

[2] *System Drivers – International Technology Roadmap for Semiconductors (2007 Edition)* [Online]. Available: http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_SystemDrivers.pdf

[3] H. Foll. *Semiconductors I – 3.3.2 Scaling Laws* [Online]. Available: http://www.tf.uni-kiel.de/matwis/amat/semi_en/kap_3/illustr/scaling.gif.

[4] *International Technology Roadmap for Semiconductors* [Online]. Available: http://www.itrs.net/reports.html

[5] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos, "Modeling Within-Die Spatial Correlation Effects for Process-Design Co-Optimization", in *Proc. Int. Symp. on Quality Electronic Design*, pp. 516–521, March 2005.

[6] R. Singhal, A. Balijepalli, A. Subramaniam, F. Liu, S. Nassif, and Y. Cao, "Modeling and Analysis of Non-Rectangular Gate for Post-Lithography Circuit Simulation," in *Proc. 44th Annu. Conf. on Design Automation*, pp. 823–828, June 2007.

[7] M. Orshansky, L. Milor, and C. Hu, "Characterization of Spatial Intrafield Gate CD Variability, Its Impact on Circuit Performance, and Spatial Mask-Level Correction," *IEEE Trans. on Semiconductor Manufacturing*, vol. 17, no. 1, pp. 2–11, Feb. 2004.

[8] M. Palusinski, A. J. Strojwas, and W. Maly, "Regularity in Physical Design," in *GSRC Workshop*, June 2001.

[9] A. J. Strojwas, "Process-Design Interaction Modeling Based Design for Manufacturability," Tutorial, in *Proc. 40th Annu. Conf. on Design Automation*, June 2003.

[10] V. Kheterpal, V. Rovner, T. G. Hersan, D. Motian, Y. Takegawa, A. J. Strojwas, and L. Pileggi, "Design Methodology for IC Manufacturability Based on Regular Logic-Bricks," in *Proc. 42nd Annu. Conf. on Design Automation*, June 2005.

[11] M. D. Stewart, G. M. Schmid, D. L. Goldfarb, M. Angelopoulos, and C. G. Willson, "Diffusion Induced Line Edge Roughness," in *Proc. SPIE*, vol. 5039, pp. 415–422, July 2003.

[12] H. Tuinhout, F. Widdershoven, P. Stolk, J. Schmitz, B. Dirks, K. van der Tak, P. Bancken, and J. Politiek, "Impact of Ion Implantation Statistics on V_T Fluctuations in MOSFETs: Comparison between Decaborane and Boron Channel Implants," in *Symp. on VLSI Technology*, pp. 134–135, 2000.

[13] Y. Li, S. M. Yu, and H. M. Chen, "Process-variation- and Random-dopants-induced Threshold Voltage Fluctuations in Nanoscale CMOS and SOI Devices," *Microelectronic Engineering*, vol. 84, pp. 2117–2120, Sept. 2007.

[14] A. Asenov, "Random Dopant Induced Threshold Voltage Lowering and Fluctuations in Sub-0.1 μm MOSFETs: A 3-D 'Atomistic' Simulation Study," *IEEE Trans. on Electron Devices*, vol. 45, no. 12, pp. 2505–2513, Dec. 1998.

[15] C. Millar, D. Reid, G. Roy, S. Roy, and A. Asenov, "Accurate Statistical Description of Random Dopant-Induced Threshold Voltage Variability," *IEEE Electron Device Letters*, vol. 29, no. 8, pp. 946–948, Aug. 2008.

[16] P. A. Stolk, F. P. Widdershoven, and D. B. M. Klaassen, "Modeling Statistical Dopant Fluctuations in MOS Transistors," *IEEE Trans. on Electron Devices*, vol. 45, no. 9, pp. 1960–1971, Sept. 1998.

[17] S. Wolf, *Silicon Processing for the VLSI Era*, Lattice Press, 1995, p. 273.

[18] A. Chandrakasan, W. J. Bowhill, and F. Fox, *Design of High-Performance Microprocessor Circuits*, IEEE press, 2001, p. 49.

[19] IBM Microelectronics. *Power Supply Trends in ASIC Products* [Online]. Available: http://www.apec-conf.org/2004/APEC04_SP1-3_IBM.pdf

[20] L. Wei, Z. Chen, M. Johnson, and K. Roy, "Design and Optimization of Low Voltage High Performance Dual Threshold CMOS Circuits," in *Proc. 35th Annu. Conf. on Design Automation*, pp. 489–494, June 1998.

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

[22] S. Sirichotiyakul, T. Edwards, C. Oh, R. Panda, and D. Blaauw, "Duet: An Accurate Leakage Estimation and Optimization Tool for Dual-Vt Circuits," *IEEE Trans. on VLSI Systems*, vol. 10, no. 2, pp. 79–90, April 2002.

[23] *Front End Processes – International Technology Roadmap for Semiconductors (2007 Edition)* [Online]. Available: http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_FEP.pdf

[24] A. Asenov, S. Kaya, and J. H. Davies, "Intrinsic Threshold Voltage Fluctuations in Decanano MOSFETs Due to Local Oxide Thickness Variations," *IEEE Trans. on Electron Devices*, vol. 49, no. 1, Jan. 2002.

[25] V. Mehrotra, S. Nassif, D. Boning, and J. Chung, "Modeling the Effects of Manufacturing Variation on High-Speed Microprocessor Interconnect Performance," in *IEEE Int'l Electron Devices Meeting Technical Digest*, pp. 767–770, Dec. 1998.

[26] Y. Chen, A. B. Kahng, G. Robins, and A. Zelikovsky, "Area Fill Synthesis for Uniform Layout Density," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 21, no. 10, pp. 1132–1147, Oct. 2002.

[27] Y. Chen, A. B. Kahng, G. Robins, A. Zelikovsky, and Y. Zheng, "Compressible Area Fill Synthesis," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 24, no. 8, pp. 1169–1187, Aug. 2005.

[28] T. Miyashita et al., "High-Performance and Low-Power Bulk Logic Platform Utilizing FET Specific Multiple-Stressors with Highly Enhanced Strain and Full-Porous Low-k Interconnects for 45-nm CMOS Technology," in *IEEE Int'l Electron Devices Meeting*, pp. 251–254, Dec. 2007.

[29] V. Chan et al., "Strain for CMOS Performance Improvement," in *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 667–674, Sept. 2005.

[30] H. S. Yang et al., "Dual Stress Liner for High Performance sub-45nm Gate Length SOI CMOS Manufacturing," in *IEEE Int'l Electron Devices Meeting Technical Digest*, pp. 1075–1077, Dec. 2004.

[31] T. Ghani et al., "A 90nm High Volume Manufacturing Logic Technology Featuring Novel 45nm Gate Length Strained Silicon CMOS Transistors," in *IEEE Int'l Electron Devices Meeting Technical Digest*, pp. 11.6.1–11.6.3, Dec. 2003.

[32] M. Yang et al., "Hybrid-Orientation Technology (HOT): Opportunities and Challenges," *IEEE Trans. on Electron Devices*, vol. 53, no. 5, pp. 965–978, May 2006.

[33] K. Su et al., "A Scaleable Model for STI Mechanical Stress Effect on Layout Dependence of MOS Electrical Characteristics," in *Proc. Custom Integrated Circuits Conf.*, pp. 245–248, Sept. 2003.

[34] V. Moroz et al., “The Impact of Layout on Stress-Enhanced Transistor Performance,” in *Int’l Conf. on Simulation of Semiconductor Processes and Devices*, pp. 143–146, Sept. 2005.

[35] M. Miyamoto, H. Ohta, Y. Kumagai, Y. Sonobe, K. Ishibashi, and Y. Tainaka, “Impact of Reducing STI-Induced Stress on Layout Dependence of MOSFET Characteristics,” *IEEE Trans. on Electron Devices*, vol. 51, no. 3, pp. 440–443, March 2004.

[36] A. Chakraborty, S. Shi, and D. Pan, “Layout Level Timing Optimization by Leveraging Active Area Dependent Mobility of Strained-Silicon Devices,” in *Design, Automation, and Test in Europe*, pp. 849–855, March 2008.

[37] R.B. Hitchcock Sr., “Timing Verification and the Timing Analysis Program,” in *Proc. 19th Annu. Conf. on Design Automation*, pp. 594–604, June 1982.

[38] A. Agarwal, D. Blaauw, V. Zolotov, S. Sundareswaran, M. Zhao, K. Gala, and R. Panda, “Statistical delay computation considering spatial correlations,” in *Proc. 2003 Conf. on Asia South Pacific Design Automation*, pp. 271–276, 2003.

[39] H. Chang and S.S. Sapatnekar, “Statistical Timing Analysis Under Spatial Correlations,” *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 24, no. 9, pp. 1467–1482, Sept. 2005.

[40] J.J. Liou, K. T. Cheng, S. Kundu, and A. Krstic, “Fast Statistical Timing Analysis By Probabilistic Even Propagation,” in *Proc. 38th Annu. Conf. on Design Automation*, pp. 661–666, June 2001.

[41] H. Chang, V. Zolotov, S. Narayan, and C. Visweswariah, “Parameterized Block-Based Statistical Timing Analysis with non-Gaussian Parameters, Nonlinear Delay Functions,” in *Proc. 42nd Annu. Conf. on Design Automation*, pp. 71–76, June 2005.

[42] M. Berkelaar, “Statistical Delay Calculation, a Linear Time Method,” in *Proc. TAU*, Dec. 1997.

[43] A. Devgan and C. Kashyap, “Block-Based Static Timing Analysis with Uncertainty,” in *Proc. IEEE/ACM Int. Conf. on Computer-Aided Design*, pp. 607–614, Nov. 2003.

[44] J. Cain and C.J. Spanos, “Electrical Linewidth Metrology for Systematic CD Variation Characterization and Causal Analysis,” in *Proc. SPIE*, vol. 5038, pp. 350–361, 2003.

[45] K. Cao, J. Hu, and S. Dobre, “Standard Cell Characterization Considering Lithography Induced Variations,” in *Proc. 43rd Annu. Conf. on Design Automation*, pp. 801–804, July 2006.

[46] P. Gupta, A. Kahng, Y. Kim, S. Shah, and D. Sylvester, "Modeling of Non-Uniform Device Geometries for Post-Lithography Circuit Analysis," in *Proc. SPIE*, vol. 6156, pp. 285–294, April 2006.

[47] M. Choi and L. Milor, "Impact on Circuit Performance of Deterministic Within-Die Variation in Nanoscale Semiconductor Manufacturing," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 25, no. 7, pp. 1350–1367, July 2006.

[48] C. Visweswariah et al., "First-Order Incremental Block-Based Statistical Timing Analysis," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 25, Issue 10, pp. 2170–2180, Oct. 2006.

[49] A. Pawloski, et al., "Line Edge Roughness and Intrinsic Bias for Two Methacrylate Polymer Resist Systems," in *J. Microlithography, Microfabrication, and Microsystems*, vol. 5, no. 2, pp. 023001, May 2006.

[50] T. Yamaguchi and H. Namatsu, "Generation mechanism of surface roughness in resists: free volume effect on surface roughness," in *Proc. SPIE*, vol. 4690, pp. 921–928, July 2002.

[51] I.T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, New York, 1986.

[52] F. Andrieu et al., "Experimental and Comparative Investigation of Low and High Field Transport in Substrate- and Process-Induced Strained Nanoscale MOSFETs," in *Proc. Symp. on VLSI Technology*, pp. 176–177, June 2005.

[53] K. Mistry et al., "Delaying Forever: Uniaxial Strained Silicon Transistors in a 90nm CMOS Technology," in *Proc. Symp. on VLSI Technology*, pp. 50–51, June 2004.

[54] D. Sylvester and A. Srivastava, "Computer-Aided Design for Low-Power Robust Computing in Nanoscale CMOS," *Proc. IEEE*, vol. 95, no. 3, pp. 507–529, March 2007.

[55] R. A. Bianchi, G. Bouche, and O. Roux-dit-Buisson, "Accurate Modeling of Trench Isolation Induced Mechanical Stress Effects on MOSFET Electrical Performance," in *Proc. Int'l Electron Devices M*eeting, pp. 117–120, 2002.

[56] A. Kahng, P. Sharma, and R.O. Topaloglu, "Chip Optimization Through STI-Stress-Aware Placement Perturbations and Fill Insertion," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 27, no. 7, pp. 1241–1252, July 2008.

[57] Y.M. Sheu et al., "Modeling Mechanical Stress Effect on Dopant Diffusion in Scaled MOSFETs," in *IEEE Trans. on Electron Devices*, vol. 52, no. 1, pp. 30–38, Jan. 2005.

[58] M. V. Dunga, C. H. Lin, X. Xi, D. D. Lu, A. M. Niknejad, and C. Hu, "Modeling Advanced FET Technology in a Compact Model," *IEEE Trans. on Electron Devices*, vol. 53, pp. 1971–1978, Sept. 2006.

[59] G. Eneman et al., "Layout Impact on the Performance of a Locally Strained PMOSFET," in *Proc. of Symp. on VLSI Technology*, pp. 22–23, June 2005.

[60] L. T. Pang et al., "Measurement and Analysis of Variability in 45 nm Strained-Si CMOS Technology," in *IEEE Journal of Solid-State Circuits*, Vol. 44, pp. 2233-2243, Aug. 2009.

[61] W. H. Lee et al., "High Performance 65 nm SOI Technology with Enhanced Transistor Strain and Advanced-low-K BEOL," in *IEEE Int'l Electron Devices Meeting*, Dec. 2005.

[62] G. Scott et al., "NMOS Drive Current Reduction Caused by Transistor Layout and Trench Isolation Induced Stress," in *IEEE Int'l Electron Devices Meeting*, pp. 827–830, 1999.

[63] Z. Luo et al., "Design of High Performance PFETs with Strained Si Channel and Laser Anneal," in *IEEE Int'l Electron Devices Meeting*, pp. 489–492, Dec. 2005.

[64] K. Ota et al., "Novel Locally Strained Channel Technique for High Performance 55nm CMOS," in *IEEE Int'l Electron Devices Meeting*, pp. 27–30, 2002.

[65] Manual, Davinci 3D TCAD, Version 2005.10.

[66] Manual, Synopsys TSUPREM4, Version 2007.03.

[67] A. Eiho et al.,"Management of Power and Performance with Stress Memorization Technique for 45nm CMOS," in *Proc. IEEE Symp. on VLSI Technology*, pp. 218-219, June 2007.

[68] T. B. Hook et al, "Lateral Ion Implant Straggle and Mask Proximity Effect," *IEEE Trans. on Electron Devices*, vol.50, pp.1946–1951, Sept. 2003.

[69] Y.M. Sheu et al., "Modeling the Well-Edge Proximity Effect in Highly Scaled MOSFETs," *IEEE Trans. on Electron Devices*, vol. 53, pp. 2792–2798, Nov. 2006.

[70] Manual, BSIM4 Spice Model, Version 4.6.1, pp. 115–116.

[71] A. Dharchoudhury et al., "Transistor-Level Sizing and Timing Verification of Domino Circuits in the PowerPC™ Microprocessor," in *Proc. IEEE Int'l Conf. on Computer Design*, pp. 143–148, Oct. 1997.

[72] T. Hori, "Gate Dielectrics and MOS ULSI," New York: Springer-Verlag, 1997.

[73] V. Zyuban and P. Strenski, "Unified Methodology for Resolving Power–Performance Tradeoffs at the Microarchitectural and Circuit Levels," in *Proc. Int'l Symp. on Low Power Electronics and Design*, pp. 166–171, Aug. 2002.

[74] S. M. Burns, M. Ketkar, N. Menezes, K. A. Bowman, J. W. Tschanz, and V. De, "Comparative Analysis of Conventional and Statistical Design Techniques," in *Proc. 44th Annu. Conf. on Design Automation*, pp. 238–243, June 2007.

[75] H. Dadgour, V. De, and K. Banerjee, "Statistical Modeling of Metal-Gate Work-Function Variability in Emerging Device Technologies and Implications for Circuit Design," in *Proc. IEEE/ACM Int. Conf. on Computer-Aided Design*, pp. 270–277, Nov. 2008.

[76] N. Sirisantana, L. Wei, and K. Roy, "High-Performance Low-Power CMOS Circuits using Multiple Channel Length and Multiple Oxide Thickness," in *Proc. Int. Conf. on Computer Design*, pp. 227–232, Sept. 2000.

[77] P. Gupta, A.B. Kahng, P. Sharma, and D. Sylvester, "Gate-Length Biasing for Runtime Leakage Control," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 25, no. 8, pp. 1475–1485, Aug. 2006.