UNIVERSITY OF CALIFORNIA, SAN DIEGO

Manufacturing-Aware Physical Design Techniques

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in
Electrical Engineering (Computer Engineering)

by

Puneet Sharma

Committee in charge:

Professor Andrew B. Kahng, Chair
Professor Chung-Kuan Cheng
Professor Rajesh Gupta
Professor Tajana Simunic Rosing
Professor Yuan Taur

2007

The dissertation of Puneet Sharma is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2007

To my loving parents, without whose support, encouragement, and sacrifices, this thesis would not have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my parents (my mother Kamlesh Sharma and my father Vijay Vinod Sharma) for their unconditional love and sacrifices. For their guidance and wisdom, without which I would not have been what I am. For their encouragement and support, without which I could not have gotten where I am today. I would like to especially thank my younger brother Hemant Sharma for his love and encouragement. For being my best friend, and for looking up to me for advice and inspiration.

My inexpressible gratitude goes to my advisor Prof. Andrew B. Kahng for his mentorship that has gone far and beyond my research area. For being not just the best advisor I could have desired, but also the expert, visionary, innovator, and enthusiastic researcher that has impacted me in several ways. I would also like to thank him for the research freedom, conducive environment, and financial support that have made my Ph.D. an enjoyable and smooth experience. Without him, I probably would not have pursued my Ph.D. and graduated with a Master's.

I would like to thank my mentor at Freescale Semiconductor, Dr. Kamal Khouri, for providing me my most enjoyable work environment, the encouragement, and the guidance. I am also indebted to him for his instrumental role in the excellent full-time opportunity that has been offered to me by Freescale. My experience at Blaze DFM was one of the most rewarding I have had, and I would like to especially thank Dr. Cho Moon, Venki Venkatesh, and Dr. Sam Nakagawa for providing me with their knowledge and perspectives.

I feel privileged to have a great group of friends and collaborators. I would like to thank Swamy Muddu, Prof. Sherief Reda, Prof. Puneet Gupta, Dr. Saumil Shah, Kambiz Samadi, Chul-Hong Park, Rasit Topaloglu, and Kwangok Jeong for their nicety, cooperation, and excellent research ideas. I am grateful to Prof. Dennis Sylvester, Prof. Alex Zelikovsky, and Prof. Ion Măndoiu for the insightful discussions and their guidance. I would like to thank Prof. Rakesh Kumar, Dr. Satya Mallick, Dr. Angshuman Parashar, Aseem Gupta, and Hrishikesh Gupta for all the great time we spent together.

I am thankful to my thesis committee members, Prof. Chung-Kuan Cheng, Prof. Rajesh K. Gupta, Prof. Tajana S. Rosing, and Prof. Yuan Taur, for taking time out of their schedules to review my research and provide useful feedback.

I would like to thank our lab's administrator Virginia McIlwain, graduate program coordinator Karol Previte, and payroll manager M'Lissa Michelson, for their support and cooperation which, at times, have gone beyond their job responsibility.

Last, but not least, I would like to express my hearty gratitude to Samiksha Sorte for her constant love and encouragement. For her sacrifices, cooperation, and levelheaded attitude which facilitated my smooth and relaxed Ph.D. For being my inspiration and the reason to succeed.

The material in this thesis is based on the following publications.

- Chapter II is based on the following publications:
  - A. B. Kahng, S. Muddu and P. Sharma, "Defocus-Aware Leakage Estimation and Control," to appear in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.
  - A. B. Kahng, S. Muddu and P. Sharma, "Defocus-Aware Leakage Estimation and Control," *Proc. International Symposium on Low Power Electronics and Design*, 2005, pp. 263 – 268.
  - A. B. Kahng, C.-H. Park, P. Sharma and Q. Wang, "Lens Aberration Aware Timing-Driven Placement," *Proc. Design Automation and Testing in Europe*, 2006, pp. 890 – 895.
  - A. B. Kahng, S. Muddu and P. Sharma, "Detailed Placement for Leakage Reduction using Systematic Through-Pitch Variation," *Proc. International Symposium on Low Power Electronics and Design*, 2007, pp. 110 – 115.
- Chapter III is based on the following publication: A. B. Kahng, P. Sharma

and R. O. Topaloglu, "Exploiting STI Stress for Performance," *Proc. International Conference on Computer-Aided Design*, 2007, to appear.

- Chapter IV is based on the following publications:
  - P. Gupta, A. B. Kahng, P. Sharma and D. Sylvester, "Gate-Length Biasing for Runtime Leakage Control," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25(8), 2006, pp. 1475 – 1485.
  - P. Gupta, A. B. Kahng, P. Sharma and D. Sylvester, "Selective Gate-Length Biasing for Cost-Effective Runtime Leakage Control," *Proc. Design Automation Conference*, 2004, pp. 327 – 330.
  - A. B. Kahng, S. Muddu and P. Sharma, "Impact of Gate-Length Biasing on Threshold-Voltage Selection," *Proc. International Symposium on Quality Electronic Design*, 2006, pp. 747 – 754.
- Chapter V is based on the following publications:
  - A. B. Kahng, P. Sharma and A. Zelikovsky, "Fill for Shallow Trench Isolation CMP," *Proc. International Conference on Computer-Aided Design*, 2006, pp. 661 – 668.
  - A. B. Kahng, K. Samadi and P. Sharma, "Study of Floating Fill Impact on Interconnect Capacitance," *Proc. International Symposium on Quality Electronic Design*, 2006, pp. 691 – 696.

My coauthors (Prof. Puneet Gupta, Prof. Andrew B. Kahng, Prof. Ion Mandoiu, Swamy Muddu, Chul-Hong Park, Kambiz Samadi, Prof. Dennis Sylvester, Rasit O. Topaloglu, Dr. Qinke (Eric) Wang, and Prof. Alex Zelikovsky) have all kindly approved the inclusion of the aforementioned publications in my thesis.

VITA

1980	Born, Delhi, India
2002	B.Tech., Computer Science and Engineering, Indian Institute of Technology, Delhi, India
2005	M.S., Electrical Engineering (Computer Engineering), University of California, San Diego
2006	C.Phil., Electrical Engineering (Computer Engineering), University of California, San Diego
2007	Ph.D., Electrical Engineering (Computer Engineering), University of California, San Diego

SELECTED PUBLICATIONS

All papers coauthored with my advisor Prof. Andrew B. Kahng have authors listed in alphabetical order.

- A. B. Kahng, S. Muddu and P. Sharma, "Defocus-Aware Leakage Estimation and Control," to appear in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

- P. Gupta, A. B. Kahng, P. Sharma and D. Sylvester, "Gate-Length Biasing for Runtime-Leakage Control," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25(8), 2006, pp. 1475 – 1485.

- P. Gupta, A. B. Kahng, I. I. Mandoiu and P. Sharma, "Layout-Aware Scan Chain Synthesis for Improved Path Delay Fault Coverage," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol 24(7), 2005, pp. 1104 – 1114.

- A. B. Kahng, P. Sharma and R. O. Topaloglu, "Exploiting STI Stress for Performance," *Proc. International Conference on Computer-Aided Design*, 2007, to appear.

- A. B. Kahng, S. Muddu and P. Sharma, "Detailed Placement for Leakage Reduction using Systematic Through-Pitch Variation," *Proc. International Symposium on Low Power Electronics and Design*, 2007, pp. 110 – 115.

- A. B. Kahng, S. Reda and P. Sharma, "On-Line Adjustable Buffering for Runtime Power Reduction," *Proc. International Symposium on Quality Electronic Design*, 2007, pp. 550 – 555.

- A. B. Kahng, P. Sharma and A. Zelikovsky, "Fill for Shallow Trench Isolation CMP," *Proc. International Conference on Computer-Aided Design*, 2006, pp. 661 – 668.

- A. B. Kahng, S. Muddu and P. Sharma, "Impact of Gate-Length Biasing on Threshold-Voltage Selection," *Proc. International Symposium on Quality Electronic Design*, 2006, pp. 747 – 754.

- A. B. Kahng, K. Samadi and P. Sharma, "Study of Floating Fill Impact on Interconnect Capacitance," *Proc. International Symposium on Quality Electronic Design*, 2006, pp. 691 – 696.

- A. B. Kahng, C.-H. Park, P. Sharma and Q. Wang, "Lens Aberration-Aware Timing-Driven Placement," *Proc. Design Automation and Testing in Europe*, 2006, pp. 890 – 895.

- P. Gupta, A. B. Kahng, S. Nakagawa, S. Shah and P. Sharma, "Lithography Simulation-Based Full-Chip Design Analyses," *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2006, pp. 61560T-1 – 61560T-8.

- A. B. Kahng, S. Muddu and P. Sharma, "Defocus-Aware Leakage Estimation and Control," *Proc. International Symposium on Low Power Electronics and Design*, 2005, pp. 263 – 268.

- P. Gupta, A. B. Kahng, P. Sharma, "A Practical Transistor-Level Dual

Threshold Voltage Assignment Methodology," *Proc. International Symposium on Quality Electronic Design*, 2005, pp. 421 – 426.

- P. Gupta, A. B. Kahng, C.-H. Park, P. Sharma, D. Sylvester and J. Yang, "Joining the Design and Mask Flows for Better and Cheaper Masks," *Proc. 24th BACUS Symposium on Photomask Technology and Management*, 2004, pp. 318 – 329.

- P. Gupta, A. B. Kahng, P. Sharma and D. Sylvester, "Selective Gate-Length Biasing for Cost-Effective Runtime Leakage Control," *Proc. Design Automation Conference*, 2004, pp. 327 – 330.

- P. Gupta, A. B. Kahng, I. Mandoiu and P. Sharma, "Layout-Aware Scan Chain Synthesis for Improved Path Delay Fault Coverage," *Proc. International Conference on Computer Aided Design*, 2003, pp. 754 – 759.

ABSTRACT OF THE DISSERTATION

Manufacturing-Aware Physical Design Techniques

by

Puneet Sharma
Doctor of Philosophy in Electrical Engineering (Computer Engineering)
University of California, San Diego, 2007
Professor Andrew B. Kahng, Chair

CMOS scaling has outpaced manufacturing technology advancements, and consequently process variability continues to increase. Manufacturing non-idealities induce variations in lateral dimensions and topography, stress variations, and material variations. These are manifested as circuit delay and power variations, and consequently low parametric yield, which is the percentage of chips that, though functional, fail to meet delay and power specifications.

Design for manufacturing (DFM) refers to measures taken during design to enhance yield. Traditional DFM techniques are essentially geometric operations with limited electrical interactions or awareness. These include resolution enhancement techniques to improve fidelity of optical lithography, design rule checks to restrict the use of layout patterns not amenable to manufacturing, and guardbanding to keep margins for process variability in design. As the extent and complexity of process variations increases, and suboptimality due to conservative design threatens to offset the benefits of scaling, these traditional DFM techniques, while still crucial, are no longer adequate.

DFM techniques to improve parametric yield can be classified according to their approach. A considerable fraction of variability is systematic in nature and can be predicted using layout and process knowledge. Examples of such variations are pitch-dependent lithography variations and layout-dependent stress effects. These variations can be predicted and compensated for in physical design to improve yield. A second class of DFM techniques enhances design robustness to

process variations. Examples include gate length biasing and redundant link insertion in clock trees, which respectively reduce leakage and clock skew variations even when the gate length variability remains the same. A third class of parametric yield-directed DFM techniques reduces process variations themselves, and includes dummy fill insertion and the increased use of layout pattern regularity.

In this thesis we propose novel DFM techniques that explicitly target parametric yield. We present three techniques for analysis and optimization of circuit leakage and delay that are knowledgeable of systematic lithography variations due to pitch, defocus, and lens aberration. Stress variations, due to width of shallow trench isolation (STI) wells, can lead to considerable delay variations. We propose timing analysis and optimization methodologies to account for STI width-dependent stress, which is highly systematic in nature. Variations in gate length arising from a variety of process variations are a major cause of leakage variability, an important problem being faced by the designers today. We propose gate length biasing, which leverages the threshold voltage roll-off to significantly reduce leakage and its variability. The technique is non-obtrusive to existing flows, easy to adopt, and inexpensive to manufacture. We also present our contributions to front-end of the line (FEOL) and back-end of the line (BEOL) fill. Our FEOL insertion methodology considerably improves topography after chemical mechanical polishing for STI and may avoid the need for reverse-etchback process steps. In BEOL fill insertion, a primary concern is the capacitive impact of inserted fill and the corresponding increase of delay and crosstalk. We describe a systematic study of the capacitive impact of inserted fill, and develop guidelines that reduce capacitive impact without sacrificing metal density.

I

Introduction

CMOS device scaling has outpaced advancements in manufacturing technology. Thus, process variability, as a fraction of feature size, continues to increase. The impact of process variations on circuit power and performance is exacerbated by the superlinear dependence of several electrical metrics on feature size (e.g., subthreshold leakage on gate length, and gate tunneling leakage on gate oxide thickness). Power, and especially leakage power, is another major challenge faced by designers today. Lowering of supply voltage to reduce dynamic power necessitates lowering of threshold voltage to sustain high-performance and adequate noise margins. Unfortunately, lowering threshold voltage causes a near-exponential increase in leakage power, and a larger ratio of static ("wasted") power to total power. Leakage variability, which is increasingly a determinant of parametric yield, is another important problem that must be addressed for continued CMOS scaling.

Traditionally, design and manufacturing have been conveniently kept separated, with only minimal information exchange. From the manufacturing side, SPICE models, technology file, and design rules are supplied for performance and power estimation, and to convey manufacturing limitations. However, in today's era of large process variability, traditional corner-based analyses can be overly pessimistic, causing valuable performance to be left on the table. Design rules also become extremely complex, substantially reducing productivity. From the design side, the layout is transferred to manufacturing as a set of shapes to be printed

on silicon. To achieve high fidelity of silicon shapes to "drawn" shapes, the manufacturing side applies several resolution enhancement techniques (RETs) to the entire design. Unfortunately, RETs significantly increase the mask writing cost and multi-million dollar mask sets are now common. To reduce mask cost, it is important to additionally convey the design intent to manufacturing so that high fidelity is attempted only for selected features in the design that require accurate manufacturing. Design for manufacturing (DFM) techniques essentially address the questions related to the exchange of information across design and manufacturing, and the use of this information for yield enhancement.

The focus of this thesis is on manufacturing-aware physical design techniques. Physical design optimizations can potentially increase yield by:

1. making the design account for process variations (e.g., systematic variation-aware design optimization techniques discussed in Chapter II and stress-aware timing optimization discussed in Chapter III);

2. increasing the robustness to process variations (e.g., gate length biasing discussed in Chapter IV reduces leakage variability even when the process variations are unchanged); and

3. reducing the process variations themselves (e.g., fill insertion discussed in Chapter V reduces topography variation).

To better understand DFM, we next present a brief overview of optical lithography.

I.A Optical Lithography

Optical lithography, or simply lithography, is the mainstream technique to create patterns on silicon wafers. While conceptually simple, lithography has evolved into a highly sophisticated process due to precision requirements that are unmatched anywhere in modern manufacturing. Lithography involves several steps

which can be simplistically grouped into photoresist deposition, exposure, and etching.

The process begins with deposition of a thin layer that is intended to be patterned on the wafer. The thin layer is sacrificial and is used to selectively etch, dope, oxidize or deposit the underlying material.[1] The pattern on the mask is first transferred to the photoresist that is deposited over the thin layer. An etchant is then used to remove the thin layer from where it is not protected by the photoresist. We now briefly describe the major lithography steps, further details of which can be found in [116].

I.A.1 Photoresist and its Deposition

Photoresists are materials that when exposed to light undergo a photochemical reaction that changes their solubility properties to a *developer* chemical. Positive photoresists become soluble in the regions that are exposed to light while negative photoresists become soluble in the regions occluded from light. Prior to deposition of the photoresist, the wafer may optionally be treated with a chemical that promotes adhesion between the thin layer and the photoresist.

The standard method of depositing the photoresist onto the wafer is *resist spinning*. In this method, a small amount of the photoresist in liquid form is dispensed onto the center of the wafer, and the wafer is then rotated about its center at a high rate. As the wafer spins, the resist spreads radially and solidifies into a uniform solid layer over the wafer. A baking step, known as *soft bake*, in which the wafer is heated to relatively low temperature for a short period of time, is then optionally performed to further densify the photoresist. Another optional step of coating the wafer with an anti-reflective coating (ARC) is then performed to suppress the light reflections in the succeeding exposure steps.

[1] Certain underlying materials can be directly etched without the use of the thin layer.

I.A.2 Exposure

By selectively exposing the photoresist to light, a pattern can be transferred to the photoresist. This process is accomplished in lithography by imaging of the mask to transfer patterns on it to the photoresist. The mask is a thin piece of a high-quality transparent material, typically quartz, partially covered with an opaque material, typically chromium, that has been removed according to the circuit pattern using an electron-beam mask writer.

Over the years, mask writing technology has improved but has failed to keep pace with the shrinkage of feature sizes. Thus, *projection printing*, in which *projection optics* (sometimes simply known as the *lens*) are used to reduce the mask image by a reduction factor (N), is now mainstream. The projection optics are typically an array of high-quality lenses cascaded to realize the reduction factor with minimal image distortion. The reduction factor in modern optics is most commonly equal to four or five. Larger reduction factors relax the precision requirements on the mask and reduce the linewidth variations due to mask errors. However, larger reduction factors increase the size of the mask and decrease the throughput in terms of the wafer area exposed under the mask. We note that the mask is also referred to as the *reticle* in the exposure context.

The equipment used to expose the photoresist-coated wafer is known as a *wafer stepper*. In a wafer stepper, a small portion of the wafer, known as the *exposure field* or simply the *field*, is exposed under the reticle through the projection optics. The illumination is then turned off and the wafer is displaced so that a different portion of the wafer is exposed in the next step. Modern wafer steppers are extremely sophisticated, with very high stepping precision. Additionally, steppers also align the wafer to the proper position so that the projected image will precisely overlay the patterns already on the wafer from previous lithography steps.

Modern wafer steppers are of the *step-and-scan* type in which the field is partially exposed through a slit [189, 116]. The lens and the wafer are translated synchronously such that the illumination through the slit scans the field from side to side. Due to the image reduction by the projection optics, the lens must be

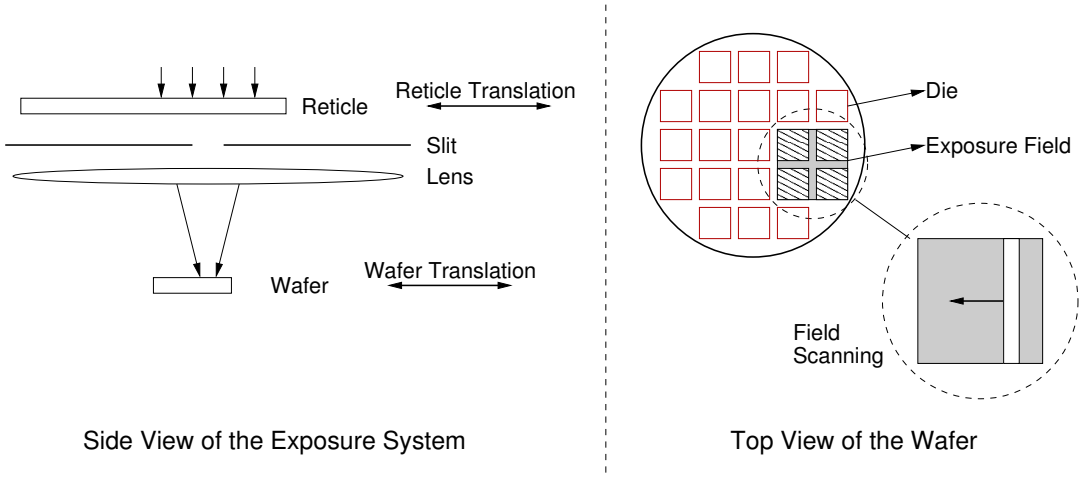


Figure I.1: Schematic of a step-and-scan wafer stepper.

translated N (i.e., the reduction factor) times faster than the wafer. Illumination through a small slit restricts the area of the projection optics that is utilized, which simplifies the projection optics and reduces their distortions. A schematic of the step-and-scan system is shown in Figure I.1.

After the patterning process completes, the photoresist undergoes *post-exposure bake*, which entails heating at a higher temperature than soft bake. The purpose of post-exposure bake is to further drive off low molecular-weight materials that may contaminate the post-lithographic equipment. Post-exposure bake also smoothes out the resist line profiles. Then, a developer solution washes away the soluble parts of the resist and the pattern has been transferred from the mask to the photoresist.

While the patterning process is highly sophisticated, the image on the mask undergoes significant distortion as it is transferred to the photoresist. Due to the extremely small sizes of the mask features, diffraction effects, inherent to the wave nature of light, become considerable. Unfortunately, a finite-sized lens is not capable of collecting all the diffraction orders as shown in Figure I.2, and the mask image cannot be completely reconstructed. This fundamentally limits the resolving capability of lithography, which is given by the following well-known Raleigh's equation:

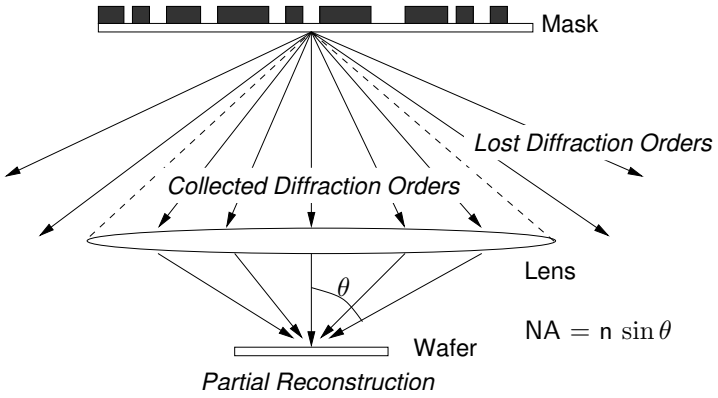


Figure I.2: Features on the mask cannot be completely reconstructed due to diffraction.

$$w_{min} = k_1 \frac{\lambda}{NA} \tag{I.1}$$

- w_{min} is the minimum feature size that can be resolved.
- λ is the wavelength of the illumination source. An ArF plasma source with a wavelength of $193nm$ is used in modern lithography, and is projected to remain in use at least through the $45nm$ node.
- NA is the *numerical aperture* of the lens and is the the sine of the maximum half-angle of light that can make it through a lens to the wafer, multiplied by the index of refraction of the medium (1.0 for air). The NA of a lens is a measure of its ability to capture the diffraction orders of light across a wide range of incidence angles.
- k_1 is known as the *k-factor* and captures the capability of the lithography process; it has a fundamental lower limit of 0.25. For modern processes, k_1 is around 0.3.

In addition to the the minimum resolvable size, the *depth of focus (DOF)* is an important parameter of a patterning system. Ideally, the wafer should be placed at the focal plane of the lens. This, in practice, is infeasible and the wafer, or certain parts of it, may be positioned at a small distance, known as the *defocus*,

from the focal plane. DOF captures the tolerance of a exposure system to defocus. DOF is given by

$$DOF = k_2 \frac{\lambda}{(NA)^2} \tag{I.2}$$

where k_2 is a constant. Similar to DOF, *exposure latitude* (ELAT) quantifies the tolerance to exposure dose variations. Together with DOF, exposure latitude gives the lithography *process window*.

Improvements in lithography equipment and resist technology, along with resolution enhancement techniques (RETs), reduce the k-factor and consequently the minimum resolvable size. RETs are methods used in lithography to enhance the printability of mask features. RETs are typically applied after signoff and before or during the mask data preparation stage. Commonly used RETs are as follows.

- *Optical proximity correction (OPC)* selectively alters the shapes of the mask patterns to compensate for patterning imperfections. OPC can be rule-based, which uses rules defined for different layout configurations, or model-based, which uses a lithography simulator. While OPC is very effective at reducing patterning variation, it requires a large runtime and significantly increases the mask complexity.

- *Off-axis illumination (OAI)* refers to illumination which has no on-axis component, i.e., which has no light that is normally incident on the mask. Examples of off-axis illumination include annular and quadrupole illumination. OAI improves the DOF for certain pitches while worsening it for others that are known as *forbidden pitches*. Fortunately, sub-resolution assist features can be inserted to eliminate or reduce the impact of the forbidden pitches.

- *Sub-resolution assist features (SRAFs)* or *scattering bars (SB)* are layout features that are inserted between layout features to improve their printability. SRAFs have very narrow widths and do not print on the wafer.

- *Phase shift mask (PSM)* adds transparent layers to the mask in certain locations to induce destructive interference at feature edges, which enhances pattern contrast and improves the k-factor.

I.A.3 Etching

Etching is used to transfer the pattern from the photoresist to the underlying thin layer. The chemical used in etching is known as the *etchant*; it selectively reacts with the underlying thin layer only in the areas that are not protected by the photoresist, while leaving the photoresist intact. The most common etching technique is reactive ion etching in which chemically reactive plasma is used to remove the thin layer in regions not protected by photoresist. After etching, the photoresist is completely removed by a variety of methods (e.g., dry etching [156]).

I.B Yield and Sources of Variability

Yield is defined as the number of chips that function *and* meet delay and power specifications, expressed as a percentage of the total number of chips manufactured. For a mature process, yield of over 90% is typical. However, during process development and ramp-up, the yield can be much less. Yield is commonly classified into the following two categories.

- *Functional yield* or *catastrophic yield* is the percentage of chips that are functional. Examples of functional failures that limit functional yield are shorts and opens in wires, open vias, line-end shortening, etc.

- *Parametric yield* is the number of chips that meet delay and power specifications, as a percentage of the functionally-correct chips. Parametric yield loss is due to chips that are functional but cannot be sold because they fail to meet the delay and power specifications.

A variety of process variations and defects cause yield loss. Functional yield loss is usually caused by misprocessing and random contaminant-related de-

fects. Parametric yield loss is typically due to process variations. However, process variations can also cause functional failures (e.g., line-end shortening leading to an always-on device) and defects can cause parametric yield loss (e.g., particle contamination that causes interconnect thinning but not a complete open).

While yield loss due to functional failures is significant, parametric failures have gained significance and now dominate functional failures. Arguably, measures to improve parametric yield are more challenging to develop and adopt. While most functional yield-enhancing methods are geometric and applied after signoff, parametric yield-enhancing methods often require understanding of the nature of process variations and modeling of their electrical effects. In this thesis, we focus on techniques that address parametric yield loss. Process variations, which are the primary cause of parametric yield loss, can be classified as follows.

- By nature – systematic vs. random. Systematic variations are predictable and can be modeled during circuit design. Random variations, on the other hand, are either unpredictable or difficult to model. Examples of systematic variations are topography variations, linewidth variation due to defocus and exposure, and stress due to shallow trench isolation (STI). Doping concentration variation, variations due to exposure system vibration, and lot-to-lot material variations are examples of random variations.

- By spatial scale – intra-die vs. inter-die. Intra-die (or within-die) variations affect the circuit components within a die differently. Examples of intra-die variations are gate length variations due to proximity effects and up to some extent topography. Intra-die variations are more difficult to account for in traditional analysis tools, and their effects are generally guardbanded. Inter-die variations include die-to-die, wafer-to-wafer, and lot-to-lot variations. They affect circuit components in a die equally and are therefore modeled as a shift in the mean.

Process variations manifest themselves as circuit metric (power and delay) variations in the following ways.

- Lateral dimension variations. The smallest dimension on a layer is referred to as the *critical dimension (CD)* because it is the most challenging to manufacture. On the polysilicon layer, CD refers to the *linewidth* of the gate poly, which is equivalent to the gate length or channel length; on metal layers, CD is the wire width. Process variations affect the CD the most, and are manifested as delay and power variations of the circuit. For example, decrease in the gate length will decrease the device delay and capacitance, but dramatically increase subthreshold leakage. Decrease in the wire width will increase resistance but decrease capacitance.

  Significant sources of CD variation are exposure and etching variations in lithography. During exposure, CD variation is due to mask errors [188], resist thickness variation [116], exposure dose variations [116], defocus [116], lens aberration [34], flare [30], etc. Microloading effects during etching also cause CD variation [83]. A substantial fraction of CD variation arising due to the these exposure and etching variations is considered systematic.

- Profile/topography variation. Chemical mechanical polishing (CMP) is performed between lithography steps to attain the designed layer height and to planarize the layer for successive process steps. Unfortunately, CMP is imperfect and cannot eliminate topography variation. Topography variation changes the metal height in back-end of the line (BEOL) which affects the wire resistance and capacitance. CMP for front-end of the line (FEOL) is used to planarize the oxide that is deposited for STI. Imperfect FEOL CMP leads to defocus during polysilicon patterning and poor inter-device isolation. Topography variation is understood to be partly systematic for both FEOL [108] and BEOL [178]. Another example is gate oxide thickness variation which affects gate-tunneling leakage and device subthreshold slope. Gate oxide is manufactured by light oxidation and its thickness variation, though small, is considered random.

- Stress effects. Mechanical stress on active regions of devices arising due to

the proximity and width of STI wells are significant in existing technologies. Stress due to STI is compressive and typically enhances the mobility of PMOS while degrading the mobility of NMOS. Consequently, delay and leakage increase for PMOS while decreasing for NMOS. Several techniques have been proposed to reduce STI stress-induced variation (e.g., [113]). STI stress is highly systematic and is partly modeled in today's design flows. Recent works have proposed modeling the residual STI stress effects [127].

- Material variations. Lot-to-lot material variations cause variations in carrier mobility, polysilicon resistance, etc. Dopant concentration in the device channels affects the threshold voltage and consequently subthreshold leakage and device delay. Due to the small number of dopant atoms in the channel in modern devices, dopant density varies significantly and randomly as a percentage and induces substantial random variation in threshold voltage.

As a consequence of these manifestations, a significant variation is seen in circuit delay and leakage. With technology scaling, process variations are increasing as a percentage, and consequently the delay and leakage variability is increasing. There is considerable parametric yield loss today especially during yield ramp-up phase causing substantial value loss.

I.C Design for Manufacturing

Design for Manufacturing (DFM) refers to measures taken during the design process to enhance yield. Parametric yield enhancement facilitated by DFM can contribute to improvement of design performance and/or power, and/or designer productivity. DFM techniques can compensate for, reduce, or make the design more robust to various types of manufacturing non-idealities.

I.C.1 Traditional Methods

While DFM has attracted great deal of attention recently from industry and academia, several techniques that can be arguably be considered DFM techniques have been in use for several years.

- RETs. As explained earlier, the purpose of these techniques is to minimize the lateral distortion between the drawn and the on-silicon shapes.

- Design rule checking (DRC). Design rules have been the primary method for the foundry to convey manufacturing limitations to design. Design rule checking, verifies adherence to these rules, and a design that is design rule-correct is expected to have a high functional and parametric yield. Simple examples of DRCs are minimum spacing, minimum and maximum dimension or area, and minimum and maximum metal density.

- Guardbanding. Considerable margin is allocated during design to account for process variations. Today's timing and power analysis flows are corner-based, i.e., a set of conservative process, voltage, and temperature (PVT) settings are assumed in analysis. With respect to process variations, hold and setup time checks are performed at fast and slow process corners respectively. Leakage is typically highest at the fast process corner, but the use of typical process corner to reduce pessimism in analysis is common. The premise behind corner-based flows is that if the design meets its specifications at conservative PVT settings, it will meet them at all other conditions. Unfortunately, this premise is not true and is now breaking down due to complex dependence of electrical metrics on variations. For example, shorter gate lengths do not necessarily have higher leakage (due to reverse short channel effect), and wider wires are not necessarily faster.

The above techniques were relatively easy to adopt and served well until the $130nm$ node. Since then, as the complexity and extent of process variations has increased, these techniques, while remaining necessary, are no longer sufficient.

Several problems stem from the inadequacy of these techniques and call for novel DFM techniques that explicitly target yield enhancement.

- With scaling, as process variations have become complex and large, design rules are no longer able to capture the variations completely and precisely. In modern technologies, layout regions that do not meet design rules may yield well, while those that meet them may not. Thus, design rules have become extremely complex in an attempt to capture process variations that arise due to complex layout configurations. Recommended design rules, which are preferably but not necessarily required to be met, have also been introduced. A large set of design rules poses maintainability problems, and limits the freedom of optimization algorithms and tools in physical design.

- The use of restricted design rules (RDRs) [118], for example those that enforce regularity by allowing only one or two pitches, increases the chip area.

- Corner-based analysis assumes conservative process conditions; this is overly pessimistic since all parameters have an extremely small likelihood of being at their conservatively assumed values at the same time. Moreover, the design metric under analysis may have a non-monotonic dependence on process parameters, in which case worst-casing the process parameter will not result in worst-casing of the design metric. An example is clock skew as a function of M3 and M4 (say) process variations. To reduce pessimism and improve worst-casing of design metrics, analysis is performed at a large number of corners. Unfortunately, the number of corners can grow rapidly with process parameters and the analysis can be both pessimistic and risky at the same time [180]. Furthermore, corner-based methods cannot account adequately for inter-die variations since all components are assumed to be at the same process corner. A notable exception is on-chip variation analysis which allows clock and data path components to be at opposite corners.

- As guardbanding increases and compromises the advantages from scaling, designers are under tremendous pressure as they seek to meet market ex-

pectations. To improve delay, power, and area of the design, considerably more time must be spent on iterations and fixing violations. This reduces productivity.

- Design rules and guardbanding can no longer be sufficiently pessimistic to ensure high parametric yield. Unexpectedly large variations and failures can cause intolerable yield loss, and require costly design re-spins.

I.C.2 Taxonomy

DFM techniques can be broadly classified into the following two categories depending on the yield loss component that they address.

- Functional yield enhancing. Several techniques have been proposed to make the design robust to random contamination-caused defects and large process variations. Critical area analysis [182] finds the chip areas that have a high chance of causing functional failures under an assumed contaminant particle size distribution. Hotspot detection [132] flags chip areas that are vulnerable to large variations due to lithography non-idealities. Examples of corresponding optimizations include wire spreading, wire widening, and via doubling. Functional yield enhancement techniques are simpler and easier to adopt because they are primarily shape-centric and have limited or no interactions with electrical metrics such as delay and power.

- Parametric yield enhancing. These techniques have attracted great interest recently as they address an ever-increasing and now dominant yield loss component. The objective of these techniques is to contain the variability in delay and leakage. This thesis focuses on such DFM techniques.

Parametric yield enhancing techniques can be further divided into the following categories.

- Design techniques to reduce process variations.

- Enhancement of design robustness to process variations.

- Systematic variation-aware design analysis and optimization.
- Statistical methods.

We now explain the four categories and give examples of each.

I.C.3 Process Variation Reduction

This category comprises techniques that reduce process variations themselves. Typically, these techniques are geometric but may have secondary design metric interactions. Examples of these techniques are:

- RETs to compensate for lithography variations;
- design rules to restrict the use of layout patterns susceptible to large variations, and the increased use of regularity; and
- FEOL and BEOL dummy fill insertion[2] to reduce topography variations.

CMP and Dummy Fill

CMP is the mainstream planarization technique used to remove excess deposited material and to attain wafar planarity over short and long ranges. CMP involves use of chemicals to soften the material to be removed, and mechanical abrasion to polish away the material. Rotary CMP tools are the most prevalent and primarily consist of a rotating *carrier* on which the wafer is mounted, and a large polishing *pad* that rotates in the same direction. The wafer is held face down and pressed against the pad. To assist polishing, *slurry*, which is a mixture of abrasive particles and chemicals that soften the material to be polished, is fed onto the pad. CMP continues until the desired thickness is attained. A common method for endpoint detection (i.e., when to stop polishing), is the use of etch-stop materials which cause the motors to draw detectably more current when the desired thickness is attained. The basic setup of rotary CMP equipment is illustrated

[2]Grobman et al. [67] group dummy fill insertion into RETs. While fill can indirectly enhance resolution (by reducing topography and hence defocus), its primary objective is to reduce topography variation.

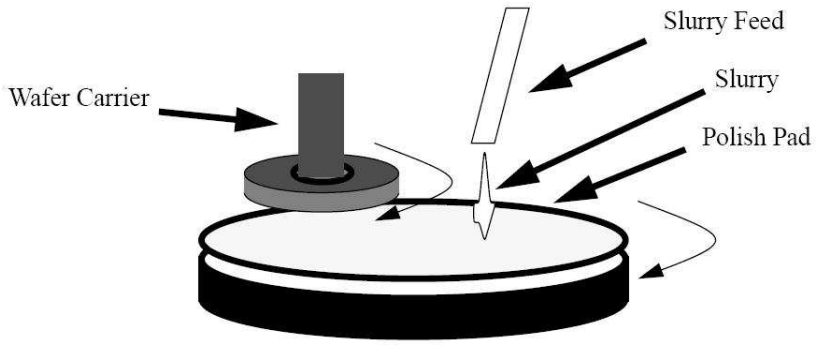


Figure I.3: Equipment used for CMP [108].

in Figure I.3. CMP is used to planarize bare wafers, in FEOL to remove and planarize overburden oxide, and in BEOL to remove excess copper and barrier, and to planarize inter-level dielectric.

While several advancements have been made in CMP technology, imperfections remain and have always been a concern due to rapidly shrinking topography variation tolerances. CMP is known to suffer from pattern-dependent problems known as *dishing* and *erosion* [169]. These two effects arise because of the existence of multiple materials of different softness that get polished simultaneously. Dishing quantifies the height difference seen in one material, while erosion captures the height loss of the harder material while polishing [178]. Two methods to reduce pattern-dependent effects are *filling* and *slotting* [96]. In fill insertion, non-functional or *dummy* geometries are added to increase the density of a material. A common objective is to make the material density over the chip uniform by adding fill to regions that have less material. Slotting works in the opposite way by removing material from large features without compromising their electrical functionality.

CMP imperfections manifest themselves into electrical variations in several ways. In FEOL, oxide dishing in STI wells and nitride erosion can cause poor isolation between devices and increase inter-device parasitics. Excessive nitride erosion into the underlying silicon, and failure to completely remove oxide from over the nitride can cause device failure. In BEOL, copper dishing and dielectric erosion affect the interconnect resistance and capacitance, and consequently the

interconnect delay. Poor planarity also poses difficulty in patterning the layers above and can cause large defocus during exposure. Planarization non-idealities also compound for higher metal layers due to the non-planarity of the underlying layer.

To achieve the tight planarity requirements, several important advancements in equipment [59], slurry [51], and topography-corrective techniques such as reverse etchback [109] have been made. However, these techniques are transparent from the design standpoint and hence outside the scope of this thesis. Dummy fill insertion and slotting are the primary design techniques used today to aid planarity by altering the density. For signal wires that are routed by gridded routers, metal density typically does not exceed $\sim 50\%$ because inter-wire spacing is nearly equal to the wire width; for these wires, slotting is not required. Slotting is done for special wires such as power/ground rails and is less desirable than fill insertion [97]. Fill insertion is the mainstream technique to increase density both for FEOL CMP [108] and BEOL CMP [178]. Fill features are rectilinear in shape that are kept separated from layout features by a separation stipulated by the design rules. While FEOL fill features are oxide features, BEOL fill is metal and can be left floating or grounded. Floating fill has the advantage that no wires have to be routed to it, while power or ground wires have to be routed to grounded fill. Floating and grounded fill differ in their effects on the capacitance of the neighboring wires.

Both floating and grounded fill are assumed to contribute identically to density increase. Most fill insertion approaches are density-driven. Density, measured over small windows of size approximately $100\mu m \times 100\mu m$, is considered a simple and reasonably accurate first-order metric for post-CMP topography. If the variation in densities computed over all such windows is small, post-CMP topography variation is expected to be small. Unfortunately, there are an infinitely-large number of such windows under continuous dissection of the design into them. So, the design is dissected discretely into windows. Density is measured in overlapping windows that are offset by a distance, known as the *tile size*, which is a fraction of the window size. A detailed description of density measurement is presented in

Chapter V. Kahng et al. [97] pointed that discrete dissection can result in underestimation of the maximum density, and proposed an efficient recursive approach to calculate the density within a user-definable error tolerance.

Density-driven fill insertion targets reduction of density variation measured over all windows. Kahng et al. [97] formalized the variation minimization problem and presented a linear programming-based density variation minimization solution. The solution finds the amount of fill that must be inserted in each tile given the capacity of each tile. To improve the scalability, a Monte-Carlo based optimization was proposed by Chen et al. [47] that iteratively inserts fill to tiles randomly selected with a probability that captures their demand for fill. Minimization of density variation is well-suited for topography variation minimization but inserts excessive amounts of fill and can dramatically change the design parasitics. Minimum fill objective, that introduces minimum amount of fill to satisfy bounded density constraints, was introduced by Tian et al. [174]. [174] associated a cost with fill insertion for each tile and proposed a formulation that minimizes the cost subject to lower and upper bounds on the *effective density*. Effective density was the density weighted by an elliptical function and its use was driven by improved CMP modeling due to Ouma et al. [138] that accounted for pad bending. While the work in [174] attempts to model pad bending effects, fill insertion that is *truly* model-based, i.e., driven by CMP simulation, remains unaddressed.[3]

FEOL fill insertion is typically rule-based and is performed by shape-based tools such as Mentor Calibre [6]. Dummy rectangles are tiled with a predefined size, spacing, and keep-off distance from the design's features. Often this approach is used to control only the nitride density along with reverse etchback which controls the oxide density. In [101], we proposed a density-driven fill insertion approach that minimizes oxide density variation and maximizes nitride density. These density objectives were identified to alleviate the failures caused by CMP and we demonstrated superior post-CMP topography metrics. Chapter V presents the details about our approach. Beckage et al. proposed a model-based

[3]Density-driven fill insertion is sometimes referred to as model-based to distinguish from fill inserted subject to only the design rules which is referred to as rule-based.

fill insertion methodology that uses CMP simulation [191, 23, 108] to identify regions for fill insertion [22]. Their approach uses two types of fill "tiles": (1) tiles that contribute to the nitride density but negligibly to the oxide density, and (2) tiles that contribute to both, oxide and nitride densities. Post-CMP topography simulation is then used to drive the insertion of these tiles in the layout.

While beneficial to the topography planarity, both FEOL fill and BEOL fill adversely affect the design through secondary effects. The capacitive increase due to BEOL fill is well-known. Floating fill increases the coupling between wires around the fill which leads to increased crosstalk. The delays of the neighboring wires could also increase especially if the wires switch in opposite directions [45]. Grounded fill, on the other hand, reduces the coupling between neighboring wires but severely increases their ground capacitance and delay. Also, methods to extract the parasitics introduced by floating fill at the full-chip scale have insufficient accuracy [102]. Thus, floating fill has to be inserted farther away which limits its ability to increase density, or grounded fill, that requires routing, has to be used. Chen et al. [46] proposed a fill insertion approach that accounts for the capacitive effects of floating fill, and the consequent impact on delays of neighboring wires. Their work utilizes simple capacitance models to insert fill for two objectives: (1) minimization of the delay impact of fill, and (2) maximization of the minimum timing slack on all wires. Recently, Xiang et al. [190] proposed a fill insertion approach to constrain the introduced coupling capacitance. Their approach budgets coupling capacitance to routing segments and uses fill-induced coupling models that are more flexible than [46] to insert fill. Grounded fill is also inserted where possible to improve robustness and predictability.

Several works have also focused on modeling of capacitive effects of floating fill. In [142] a model-library based approach to extract floating-fill was briefly described but results demonstrating the accuracy and characterization time were omitted. [111] presented a methodology for full-chip extraction of *total* capacitance in presence of floating-fill and [112] extended their analysis. Their approach adjusts the permittivity and sidewall thickness of dielectric to account for the capacitance increase due to fill so that off-the-shelf extractors can then be used. We

studied the effect of different fill shape and configuration parameters through field solver simulations [99]. We also proposed a set of guidelines for fill insertion that reduce the capacitive effect for the same density of fill inserted. Details of our work are presented in Chapter V. Topaloglu [176] presented a framework to utilize our guidelines for automatic fill insertion. Kahng et al. [102] have recently proposed a set of experiments to construct tables for floating fill extraction in full-chip tools.

FEOL fill can increase the capacitance between overlying Metal 1 wires. Additionally, capacitance can increase between an overlying Metal 1 wire and an adjacent active region of a device. More significant effect is the reduction in STI stress that occurs because inserted fill decreases the width of STI. Stress affects the carrier mobility and hence the delay and leakage of CMOS devices. Miyamoto et al. [125] analyzed the layout-dependent stress that occurs due to STI. Moronoz et al. [127] studied the effect of fill on the performance of nearby devices and also indicated the use of fill to improve performance. In [100] we proposed a delay optimization methodology that inserts fill next to NMOS devices to reduce their stress and improve performance. Chapter III describes our methodology.

Looking forward, the use of conservative minimum separation design rules to limit the design impacts of fill will be too restrictive for planarity control. Next-generation fill insertion techniques should therefore comprehend and model the design effects of fill in fill insertion. The pros and cons of integrating fill insertion within routing were presented in [98]. Ever-tightening planarity requirements also indicate that explicit CMP modeling to assess the topography consequences of fill insertion might be used in future fill insertion flows. Multi-objective fill (e.g., BEOL fill insertion to augment power distribution [115]) is also interesting.

Reducing Lithography Variations

Lateral dimension variations, specifically variations in CD, cause delay and leakage variations. In the FEOL, variation in gate polysilicon (poly) length (i.e., device gate length) and diffusion region size cause delay and leakage variations, while variations in field poly affect parasitics. In the BEOL, variations in

wire width cause variations in parasitics and consequently wire delay. CD variability is caused primarily by lithography variations (i.e., mask, imaging, and etching errors). Thus, it is important to reduce lithography variations.

RETs are the primary methods to reduce lithography variations. Typically, RETs are transparent to the design phase and are performed after signoff. However, modifications can be made to a circuit that make it more amenable to RETs, such that the RETs achieve stronger reduction of lithography variation. Gupta et al. [75] propose three flows for scattering bar and etch dummy insertion. Some pitches, especially with the use of OAI, have poor printability; these pitches are known as *forbidden pitches*. Scattering bar insertion reduces the occurrences of forbidden pitches and enhances printability. [75] proposes a dynamic programming-based detailed placement algorithm to reduce or eliminate the number of forbidden pitches. Etch dummies are non-functional geometries added to the active layer to protect devices near the active edges from ion scattering during etching. Scattering bar insertion interferes with etch dummy insertion because of specific etch dummy to scattering bar spacing rules. A scattering bar-aware etch dummy insertion flow is also proposed in [75] to make the layout more conducive to scattering bar insertion after etch dummy insertion. Finally, [75] presents a detailed placement approach for etch dummy insertion. The reported results show substantial reduction in the number of forbidden pitches, and in the edge placement error (EPE)[4] due to exposure and etch non-idealities.

Kahng et al. [93] propose the use of "auxiliary patterns" which are similar in function to scattering bars but wider and hence more effective at shielding CD from proximity effects. The disadvantage of auxiliary patterns is that, unlike scattering bars, they print on the wafer and may require whitespace for their insertion. A detailed placement approach was proposed to apply auxiliary patterns to all cells in a design with no area overhead. The approach was proposed in the context of cell-based OPC to reduce OPC runtime, but can be used to reduce CD errors that arise due to optical proximity effects (i.e., through-pitch CD variation).

[4]Edge placement error (EPE) refers to the number of CD edges for which the error exceeds a given threshold.

I.C.4 Design Robustness Enhancement

We group the techniques that improve the design robustness to process variations in this category. These techniques reduce the sensitivity of design metrics to process variations so that the design metric variability decreases even when the process variations remain the same. Functional yield enhancement techniques that address yield loss due to particle contamination utilize this approach since particle contamination is considered to be random and cannot be reduced. Examples of these techniques are wire widening, wire spreading, and via doubling. These techniques are sometimes selectively applied to regions identified by critical area analysis [182] to be highly susceptible to particle contamination-induced failures. Device layers are generally considered to be less susceptible because devices are grouped and laid out in standard cells which by construction reduce the susceptibility of device layouts. Also, modifications to the device layers change cell parasitics and require re-characterization for performance and power analysis. However, methodologies such as pDfx [9] are capable of instantiating (swapping in) high yielding cells in the design where possible. A methodology that reduces critical area by swapping in enhanced-yield cells having smaller critical area in a timing-driven manner is proposed in [87].

Wire widening increases robustness against open failures that occur when particle contamination leaves gaps in wires that prevent electrical connectivity of wire endpoints. Wire spreading increases robustness against short circuits between adjacent wires that can occur when a contaminant connects adjacent wires. Via doubling is used to enhance yield loss attributed to open vias which are caused by imperfect manufacturing (e.g., stress-induced via voiding [143]). While these yield loss mechanisms typically affect functional yield, they can also be relatively subtle and affect parametric yield. A particle contaminant may not completely disconnect the endpoints of a wire but only reduce the wire width, thus increasing the wire resistance and hence its delay. Similarly, via voiding may not cause a via to become open but only increase its resistance. Hence, techniques that enhance robustness against these failures are also considered as DFM methods that enhance

parametric yield.

One of the simplest yet among the most effective ways to increase robustness is the use of redundancy. The use of redundancy in memories in the form of error correcting codes (ECC) is well-known. Via doubling is a DFM technique that introduces via redundancy to increase robustness against open and/or high-resistance vias. Redundant link insertion in clock trees has been proposed [179, 148] to reduce clock-skew variation even when process variations on clock buffers and interconnects remain the same. These techniques insert cross links between nodes of a clock tree that have zero nominal skew to obtain a non-tree that is more robust to variations. The premise is that the variation of a clock signal that propagates through multiple paths is less than the variation of individual paths.

In Chapter IV, we propose gate length biasing that selectively increases the gate lengths of devices that are not timing-critical. Our technique significantly reduces leakage variability even when the gate length variability remains the same because of the following reason. Leakage decreases exponentially with gate length, and with larger gate lengths, the gradient of the leakage vs. gate length curve is smaller. Thus, we exploit the non-linear nature of the leakage vs. gate length curve to reduce leakage variability.

Circuit topology is well-understood to affect susceptibility to variations. A circuit with high logic depth is expected to have a smaller delay variation because cell delay variations compensate each other, up to some extent. Mathematically, the relative standard deviation of a sum of random variables is no more than the maximum relative standard deviation of the individual variables. If the delay variations of the cells on a path are assumed to be independent and distributed normally, the standard deviation is given by the square root of the sum of squares of individual standard deviations, and is smaller than the sum of the individual standard deviations. This compensating effect can be somewhat dampened by correlations. Nevertheless, increasing logic depth is an effective robustness enhancement technique.

Timing slack profile also determines the circuit's susceptibility to process variations. Specifically, a circuit with a timing slack profile that has a large number

of critical paths (i.e., with a small slack) is more susceptible to variations than one with fewer critical paths. This is because the distribution of the maximum delay shifts (e.g., $\mu+3\sigma$ metric) to a higher delay value when there are a large number of critical paths. Hence, reducing the number of critical paths increases robustness. Lowering the use of cells and circuit structures that are vulnerable to process variations also increases robustness. For example, low threshold-voltage cells are extremely susceptible to random dopant fluctuations because of their low doping density. These process variations manifest as variations in threshold voltage [117], and consequently in leakage and delay. Reducing the use of cells susceptible to variations naturally enhances robustness.

I.C.5 Systematic Variation-Aware Analysis and Optimization

This category includes measures taken during design that explicitly address systematic variations. Several of the variations are largely systematic in nature and can be accounted for in design analysis and optimization. Examples of such variations are as follows.

- CD variations arising due to imaging non-idealities such as through-pitch variation, defocus, lens aberration, flare, etc.
- Topography variations, both in FEOL and BEOL, that arise due to material density variations. Variations in topography may manifest as feature height variations as well as defocus variations in the patterning of the layer above.
- Stress variations that arise due to STI, contact placement, etc.

Variations in Photolithography Imaging

Orshansky et al. [134] studied intra-die gate length variability in a $180nm$ process and reported systematic variations to be more significant than random variations. Further, the authors observed spatially-correlated variations to exceed

context-dependent variations that arise due to proximity effects. The observed extent of gate length variations induced a 25% variation on clock cycle time and the need for a systematic variation-aware timing analysis methodology was highlighted. The authors used a simple relationship between the gate length and the cell delay, and proposed a location-dependent timing analysis flow that accounts for spatial gate length variations.

Gupta et al. [70] address the timing analysis implications of systematic variation in across-chip CD that arises due to imperfect defocus. Error in device CD (i.e., gate length) can be modeled once the defocus and pitch of the device are known. Gate delays depend on the CDs of the constituent devices, and the impact of across-chip CD variation on timing can be modeled. The timing analysis methodology proposed in [70] constructs variants of all cells in the library corresponding to different neighborhood contexts. The appropriate variant is then selected from the library on the basis of the layout context of the cell to run timing analysis. The authors report a reduction of up to 40% in the timing guardband with respect to traditional corner-based analysis.

Yang et al. [192] propose a timing analysis flow that utilizes the CDs predicted by lithography simulation to more accurately perform timing analysis. In their methodology, lithography simulation is conducted on the critical cells (as identified from a prior traditional timing analysis) to estimate the gate lengths of all the devices in them. Then, SPICE netlists of the critical cells are modified with the estimated device gate lengths, and standard-cell characterization is run. The critical cells are then mapped to the appropriate cell master in the library, and timing analysis is run. The authors report considerable change in slacks of several critical paths.

In [73], we have proposed a lithography simulation-based full-chip design analysis methodology. Our approach analyzes the entire circuit for delay, leakage, and dynamic power. The fundamental difference between our approach and [192] is that we construct a fixed number of variants in our cell library and perform mapping of each lithography simulated cell instance to the most appropriate cell master in the library. We address the complications that arise in simplifying con-

tours (that are outputted by lithography simulation) to rectilinear shapes that are suited to off-the-shelf analyses tools. Our simplification policies and methodologies to select the most appropriate cell master from the library are dependent on the analysis of interest. We also address the impact of interconnect dimensional variation on interconnect parasitics. Zhou et al. [198] propose a methodology that performs lithography simulation on the interconnects prior to parasitic extraction. The focus of their work is on construction of extraction rule decks using a 3D field solver for shapes outputted by lithography simulation.

Cao et al. [35] also propose a full-chip timing and power analysis approach based on lithography simulation. In their approach, dummy features are inserted within a cell layout along the boundaries to shield from proximity effects. If on insertion of dummy features the proximity effects can be assumed to be negligible, all cell instances in a design experience identical lithography variations. Thus no additional cell variants are needed in the library, and the cells can be characterized to account for the impact of lithography variations. The authors report $8\% - 25\%$ reduction in timing guardband and 55% reduction in power guardband with respect to traditional corner-based analyses.

Gupta et al. [72] propose a timing optimization approach that exploits the opposite lithography-induced gate length variations experienced by dense and isolated pitches to compensate for each other. In their process, gate lengths of dense devices (i.e., devices with small spacings from neighboring devices) increase with defocus, while those of isolated devices decrease. The proposed approach constructs isolated and dense variants for all cells in the library. An optimizer is then used to select either a dense or an isolated variant from the library to map each of the cell instances to. The objective of the optimizer is to use a mix of isolated and dense variants such that the delay and leakage variabilities due to defocus reduce.

A detailed placement approach to reduce CD error is proposed in [85]. Placement affects the pitches of devices in a layout, which determine CD variation arising because of proximity effects. The authors run lithography simulation to estimate the CD error when any two cells are placed next to each other. Then a

traveling salesman problem-based detailed placement optimization is used to minimize CD variation. Since detailed placement can adversely increase wirelength, a wirelength-constrained solution is also proposed. The authors report over 20% reduction in EPE.

Mitra et al. [124] propose a lithography-aware routing technique that guides an off-the-shelf router to minimize EPE. First, EPE for the layout is estimated using lithography simulation. In each routing grid cell, the cumulative EPE density is calculated, and grid cells are processed in decreasing order of their cumulative EPE density. Two routing modifications are proposed in the paper: (1) spreading of routing segments in the neighborhood of a large EPE routing segment, and (2) addition of blockages followed by ripup-and-reroute. Fast aerial image simulation is also developed to monitor the impact of routing modifications on EPE. The authors found insertion of blockages followed by ripup-and-reroute to be effective at EPE reduction and report the associated EPE reduction to be up to 40%.

Modeling and Accounting for Systematic Topography Variations

Post-CMP topography variations are believed to have a large systematic component that is predictable from the layout of the underlying layer [169]. Models for post-CMP topography simulation have been developed for FEOL by Lee [108], and for copper BEOL by Tugbawa [178]. These models, among others, can be used to drive analyses and optimizations of metrics that depend on topography.

Gupta et al. [76] propose a topography-aware OPC flow. Traditional OPC is applied for a specific user-inputted defocus value. At defocus other than the specified value, the effectiveness of OPC to control lithography variations is diminished. Topography is a significant source of defocus [63] and its simulation can partly estimate defocus. The technique of [76] performs topography simulation to predict the defocus at different regions of a design. The defocus values are then binned and passed to OPC through additional annotation layers in GDSII file format. The authors report up to 67% reduction in EPE for a $90nm$ technology

with a topography variation of $\pm 100nm$. In Section II.B, we describe our leakage analysis and optimization approach that uses topography variation.

Cho et al. [50] propose global routing that accounts for topography variation. The authors observe that interconnect height increases, and consequently its resistance decreases, as the wire density decreases. Also, the coupling and total capacitance decrease with wire density. Thus, for timing-critical nets, it is beneficial to have low wire density in their neighborhood. The proposed router essentially reduces wire density in the vicinity of timing-critical nets to improve their speed, and reduces wire density of high-density grid cells to reduce overall CMP variation. With the proposed approach, the authors claim a reduction of 8% in the minimum clock cycle time with negligible wirelength increase.

The impact of topography on interconnect parasitics is more extensively studied by He et al. [82]. Using the topography model developed for Copper CMP in [178], [82] estimates the change in parasitics with 3D field solver simulations. For a $1mm$ interconnect in $65nm$ technology, the authors report the resistance to increase by 30% when CMP-induced copper dishing is accounted for. Capacitance impact was relatively small and typically under $\pm 3\%$ for coupling capacitance, and $\pm 0.3\%$ for total capacitance. However, with the insertion of fill, coupling capacitance increases by 30% to 140%, and total capacitance is impacted by -1.35% to 1.88%. The authors also propose a dynamic programming-based simultaneous wire sizing and buffer insertion algorithm that accounts for changes in parasitics due to fill insertion and post-CMP topography. With respect to traditional buffer insertion and wire sizing that is oblivious of CMP and fill effects, the proposed approach improves delay by 1.6%.

I.C.6 Statistical Methods

Statistical analysis and optimization techniques have been applied for delay and leakage. Statistical static timing analysis (SSTA) has particularly attracted great attention [65, 19, 26, 119, 57, 18, 181, 39, 196, 41, 197, 159, 183, 160] because it addresses several limitations of traditional static timing analysis (STA). STA is

corner-based and assumes *conservative* process, voltage, and temperature (PVT) conditions which yield an overly pessimistic analysis and leave valuable power and performance improvements on the table. The number of STA runs required, especially when the designers attempts to reduce the pessimism in analysis, can be intractably large. Further, [180] describes how corner-based traditional STA can be pessimistic and risky at the same time. STA does not adequately account for intra-die variations which can be considerable. Futhermore, correlations arising because of spatially-correlated process variations and path reconvergance cannot be satisfactorily handled in STA.

SSTA is characterized by the use of random variables for analysis quantities (delays, slews, capacitance; consequently, arrival times, required times, slacks) instead of deterministic variables. SSTA, essentially, computes the probability densities associated with each of the random variables. Perhaps the simplest SSTA approach is *Monte-Carlo analysis* which involves multiple iterations of STA. In each iteration, the distributions of all process parameters are sampled and STA run for those values. The calculated timing values have a probability equal to the probability of all process parameters having their sampled values. The approach can handle general parameter distributions, parameter correlations, and arbitrary delay functions. Unfortunately, to generate useful distributions, the approach requires a large number of STA iterations and becomes intractable even for small circuits. Several more efficient SSTA algorithms have been proposed that can be broadly categorized into two classes – path-based and block-based.

Path-Based SSTA

In path-based SSTA, a set of timing paths is analyzed. The set usually comprises of the paths that are evaluated to be critical (i.e., have slack less than a certain threshold) by STA and have a reasonable probability to limit timing-yield. Gattiker et al. presented a path-based SSTA approach in which path delays are expressed as linear function of the process variations [65]. The technique was later enhanced by Agarwal et al. [19]. The primary shortcoming of path-based SSTA is

that in modern circuits, that undergo several optimization stages, the number of critical paths can be very large and path-based SSTA can often become intractable. Furthermore, Path-based SSTA is not suitable for optimizations because it does not provide metrics and diagnostics required by optimizers such as cell slack, and because it is not amenable to efficient incremental processing.

Block-Based SSTA

Block-based SSTA is similar to traditional STA in the sense that cells are processed in a topological order. Despite poorer accuracy than Monte-Carlo and path-based SSTA, block-based SSTA algorithms have attracted the most attention due to their efficiency. They are also suitable for statistical optimizations. Issues in block-based SSTA include consideration of correlations that arise due to path reconvergence and process variations, complexity of delay models as functions of process variations, assumption of probability distributions that get propagated. Correlations arising due to reconvergence and process variations were ignored in early works [26, 119]. Later [57] and [18] presented general frameworks to account for correlations due to reconvergence and spatial correlations respectively. Efficient and incremental computation techniques that are based on first-order delay models and assumed Gaussian distributions are presented in [181, 39]. Extensions that support higher-order delay models and lift assumptions about Gaussian distributed process variations at the cost of significant runtime increase are proposed in [196, 41, 197]. Recent works have focused on improving computational efficiency [159], methods that do not rely on complete availability of process variation information [183], and use of affine interval-based methods to derive tight bounds [160]. Statistical power minimization has also been the subject of several papers [167, 122].

While substantial progress has been made in statistical methods, several challenges to their adoptability remain. First, rigorous assessment of SSTA in real-world design and manufacturing has been very limited, if at all any. The value proposition is not clear; it is unclear whether the underlying assumptions and ap-

proximations in SSTA are reliable, and if a methodology that captures variations with adequate accuracy will be computationally tractable. Second, feeding statistical methods with variational data is challenging. Some issues are: assumptions about variational characteristics, test structure design and collection of variational data, and transfer of data in a standardized format that supports flexibility and confidentiality. Third, statistical methods are considerably more sophisticated and would require a steep learning curve for a large group of designers.

I.D This Thesis

The focus of this thesis is on parametric yield enhancing physical design techniques. At the physical design stage, considerable information is available about the design layout, delay, and leakage, all of which can be utilized to drive accurate analyses and effective optimizations. The presented techniques are organized into four chapters: (1) analysis and optimizations enabled by systematic lithography variations, (2) stress-aware timing analysis and optimization, (3) gate length biasing to enhance robustness to gate length variations and to reduce leakage, and (4) fill insertion for FEOL and its guidelines for BEOL.

In Chapter II, we present techniques that perform design analyses and optimizations with the knowledge of systematic lithography variations due to pitch, focus, and lens aberration. We present three techniques:

- Defocus-aware leakage analysis and optimization. A significant fraction of variation in linewidth occurs due to systematic variations involving focus and pitch. Leakage depends nearly exponentially on linewidth and prediction of linewidth can considerably improve leakage estimation. We propose a new leakage estimation methodology that accounts for focus-dependent variation in linewidth. Our approach computes the pitch of each device in the design and uses it along with defocus information to predict the linewidth of the device. Once the linewidths of the devices in a cell are calculated, the cell leakage is computed to be the sum of leakages of all off-state devices

in the cell; device leakages are found from a linewidth-leakage table that is pre-characterized with SPICE simulations. The presented methodology significantly improves leakage estimation and can be used in existing leakage reduction techniques to improve their efficacy. To demonstrate the use of our approach for leakage reduction, we modify the gate length biasing of [77] to consider systematic variations in linewidth and further optimize leakage power.

- Detailed placement to improve leakage using through-pitch variations. We present a novel detailed placement technique that accounts for systematic through-pitch variations to reduce leakage. A substantial fraction of linewidth variation is systematic with respect to the device layout context. Detailed placement changes context of the devices that are near the cell boundaries and can be used to reduce leakage. Our approach modifies the placement of cells in small windows such that contexts that reduce leakage are created. During this optimization, cells are partitioned into rows and then placed in rows using a traveling salesman problem formulation

- Aberration-aware timing analysis. Process variations due to lens aberration are to a large extent systematic, and can be modeled for purposes of analyses and optimizations in the design phase. Traditionally, variations induced by lens aberration have been considered random due to their small extent. However, as process margins reduce, and as improvements in RETs control variations due to other sources with increased efficacy, lens aberration-induced variations gain importance. We present a novel timing analysis flow, that utilizes *Zernike* coefficients that quantify aberration along with layout information, to perform a more accurate analysis and reduce timing guardband.

In Chapter III, we present a STI stress-aware timing analysis and optimization methodology. Starting at the 65nm node, stress engineering to improve performance of transistors has been a major industry focus. An intrinsic stress source, STI, has not been fully utilized up to now for circuit performance im-

provement. We present a new methodology that combines detailed placement and active-layer fill insertion to exploit STI stress for performance improvement. We start with process simulation of a production $65nm$ STI technology, and generate mobility and delay impact models for STI stress based on these simulations. Based on these models, we are able to perform STI stress-aware modeling and simulation of critical paths using SPICE. We then present our timing-driven optimization of STI stress in standard-cell designs, using detailed placement perturbation to optimize PMOS performance and active-layer fill insertion to optimize NMOS performance. The frequency improvement through exploitation of STI stress comes at practically zero cost with respect to area, wirelength and design cycle time.

Chapter IV presents our gate length biasing technique, that enhances design robustness by reducing the design leakage and its susceptibility to gate length variations. We study the additional design space afforded by biasing of device gate lengths to reduce chip leakage power and its variability. It is well known that leakage power decreases exponentially, and delay increases linearly, with increasing gate length. Thus, it is possible to increase gate length only marginally to take advantage of the exponential leakage reduction, while impairing performance only linearly. From a design flow standpoint, the use of only slight increases in gate length preserves pin- and layout-compatibility; therefore, our technique can be applied as a post-layout enhancement step. We apply gate length biasing only to those devices that do not appear in critical paths, thus assuring zero or negligible degradation in chip performance. To highlight the value of the technique, we first apply the multi-threshold voltage technique which is widely used for leakage reduction, and then use gate length biasing to show further reduction in leakage. Selective gate length biasing at the circuit level reduces circuit leakage by up to 30% in our testcases with no delay penalty. Leakage variability is reduced significantly by up to 41%, which may lead to substantial improvements in manufacturing yield and product cost.

In Chapter V, we focus on fill insertion for both FEOL and BEOL. We present two techniques:

- FEOL fill for improved planarity. STI is the mainstream CMOS isolation technology and relies on CMP to remove excess of deposited oxide and attain a planar surface for successive process steps. Despite advances in STI CMP technology, pattern dependencies cause large post-CMP topography variation that can result in functional and parametric yield loss. Fill insertion is used to reduce pattern variation and consequently decrease post-CMP topography variation. Traditional fill insertion is rule-based and is used with reverse etchback to attain desired planarization quality. Due to extra costs associated with reverse etchback, "single-step" STI CMP in which fill insertion suffices is desirable.

  To alleviate the failures caused by imperfect CMP, we focus on two objectives for fill insertion: oxide density variation minimization and nitride density maximization. A linear programming based optimization is used to calculate oxide densities that minimize oxide density variation. Next a fill insertion methodology is presented that attains the calculated oxide density while maximizing the nitride density. Averaged over the two large testcases, the oxide density variation is reduced by 63% and minimum nitride density is increased by 79% compared to tiling-based fill insertion. To assess post-CMP planarization, we run CMP simulation on the layout filled with our approach and find the planarization window (time window in which polishing can be stopped) to increase by 17% and maximum final step height (maximum difference in post-CMP oxide thickness) to decrease by 9%.

- BEOL fill insertion guidelines to reduce capacitance impact. It is well known that fill insertion adversely affects total and coupling capacitance of interconnects. While grounded fill can be extracted by full-chip extractors, floating fill can be reliably extracted by 3D field solvers only. Due to poor understanding of the impact of floating fill on capacitance, designers insert floating fill conservatively. We study the impact of floating fill insertion on coupling and total capacitance when the fill geometry, and both the interconnects between which the capacitance is measured are on the same layer. We show that the

capacitance with same-layer neighboring interconnects is a large fraction of total capacitance, and that it is significantly affected by fill geometries on the same layer. We analyze the effect of fill configuration parameters such as fill size, fill location, interconnect width, interconnect spacing, etc. and consider edge effects and effects occurring due to insertion of several fill geometries in close proximity. Based on our findings, we propose certain guidelines to achieve high metal density while having smaller impact on interconnect capacitance. Finally, we validate the proposed guidelines using representative process parameters and a 3D field solver. On average, coupling capacitance increase due to floating-fill insertion decreases by $\sim 53\%$ on using the proposed guidelines.

II

Utilizing Systematic Variations in Analysis and Optimization

II.A Introduction

Optical lithography continues to be a key enabler of the aggressive IC technology scaling implicit in Moore's Law. Feature size scaling has outpaced the improvements in lithography hardware solutions, so that linewidth tolerances are extremely difficult to achieve. Hence, RETs such as OPC, PSM, and OAI are being pushed ever closer to fundamental resolution limits.

Despite the use of RETs, substantial CD variation arises in modern technologies due to imperfections in the exposure system. Other important sources of CD variation are mask errors, mask misalignment, and microloading effects during etching. CD variation because of imperfect imaging is primarily caused by the following:

- Existence of non-ideal process conditions. RETs are tuned for a set of process conditions of defocus and exposure dose. However, variations due to topography, wafer stage errors, lens aberration, flare, lens heating, etc. cause the process conditions to deviate from ideal and result in significant CD variation.

- Restricted use of RETs. RET application is computationally expensive, and increases the feature complexity which adversely affects mask cost. Typically, a small error tolerance of $1-2\%$ in the edge placement error is given to the RET algorithm to reduce RET runtime and feature complexity. RET is also limited by the capability of mask writing technology. Thus, a residual CD error arises from the error tolerance in RET.

- Limitations of RET. Rule-based RETs perform inadequately on complex layout configurations which are not captured well by the rules. While modern RETs are model-based and significantly more sophisticated, they still have limited optical modeling, which is computationally expensive, and CD error increases with more complex layout configurations.

CD variation can be split into systematic and random components. Systematic variation can be modeled using the design and lithography information. Random variation includes unpredictable and difficult to model variations. In this chapter we limit our scope to systematic variations arising from pitch, defocus, and lens aberration.

The polysilicon layer is perhaps the layer that is most significantly affected by CD variation. On the polysilicon layer, CD refers to the linewidth of NMOS and PMOS devices and is equivalent to the gate length or channel length. Typically, the linewidth is the smallest and therefore the most challenging dimension to print in the entire design. Also, linewidth variation has the most direct impact on the device drive current, capacitance, and leakage current. Via and metal layers, while containing small features, are arguably less important from the design performance or power standpoint. In this chapter we study the effect of systematic linewidth variations on circuit delay and power, and develop systematic variation-aware analysis and optimization methodologies.

II.A.1 Systematic Through-Pitch and Through-Focus Variation

A substantial fraction of linewidth variation is systematic with pitch and defocus as shown in Figure II.1. The figure plots interpolated foundry data *after application of RETs including OPC and scattering bar insertion* [33] over a realistic defocus range for a $65nm$ technology. The quadratic dependence of linewidth on defocus has been reported and used in several previous works [121, 63, 70]. We have found linewidth for multiple devices with similar pitch and defocus to be nearly identical in foundry data. Therefore, the plot in Figure II.1 can be used to predict linewidth given pitch and defocus. From the plot we note that:

1. the linewidth of dense pitches increases with defocus while that of isolated pitches decreases;
2. dense lines have a larger linewidth than sparse lines across all defocus values;
3. sparser lines exhibit a larger linewidth decrease with defocus; and
4. the linewidth change due to pitch saturates as the pitch approaches the optical radius.

This systematic nature of across-chip linewidth variation (ACLV) has been exploited in recent works for timing analysis [70], design robustness [71], and leakage analysis and systematic-variation aware linewidth biasing [90]. Systematic ACLV of a device can be predicted once the layout pitch and defocus are known. While pitch is deterministic and is known after placement, defocus is a random variable. Fortunately, lines of different pitches have different sensitivities to defocus variation, and linewidth can be predicted to some extent based only on the pitch information. For example, dense lines will always have a linewidth larger than sparse lines as shown in Figure II.1. Furthermore, under an assumed defocus distribution, the expected linewidth value of any pitch can be calculated using Figure II.1.

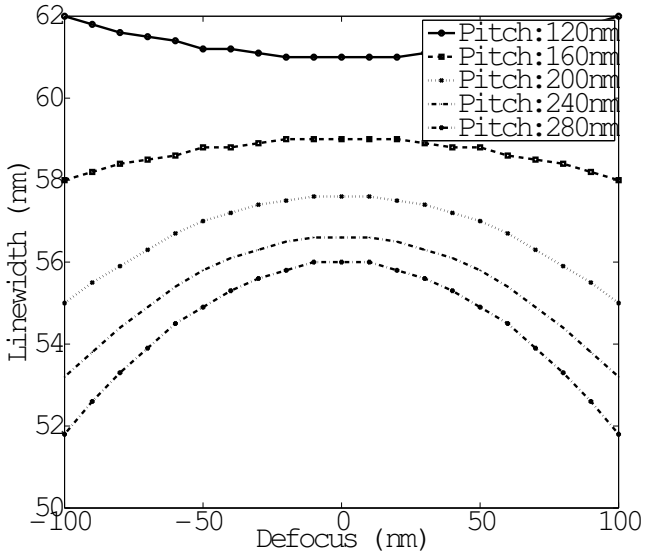


Figure II.1: Variation of simulated on-silicon linewidth with defocus for different pitches for a $65nm$ technology. Linewidth increases with defocus for dense (small pitch) patterns, and decreases for sparse (large pitch) patterns.

In certain scenarios, partial information about defocus may be available which can further enhance linewidth prediction. Defocus is caused by several sources, such as variation in STI layer thickness during CMP, lens aberration, wafer stage misalignment, and resist thickness variation [52]. Linewidth variation caused by defocus due to thickness variation can be systematically modeled from layout density analysis and physical CMP models.

A schematic of topography-dependent defocus during lithography is shown in Figure II.2. If the image plane of the reticle and lens system coincides with the wafer plane, the image prints with high resolution. However, in the regime of topography variation, caused predominantly by erosion and dishing effects during CMP, the image prints out of focus, leading to topography-dependent linewidth variation.

Topography simulation has been the focus of several recent papers, e.g., [191][108]. These works present and calibrate analytical models that account for the underlying pattern and various CMP process parameters such as planarization

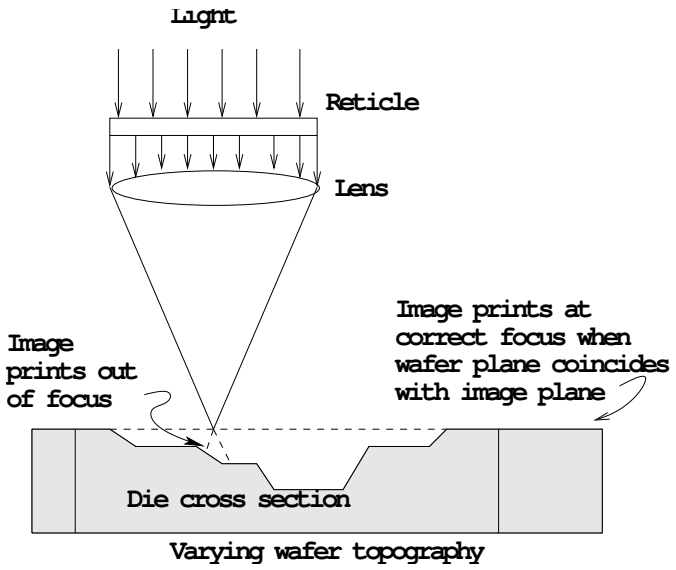


Figure II.2: A vertical cross section of a wafer showing topography non-uniformity. Focus variation due to non-planar wafer topography is illustrated.

length, pad bending, slurry selectivity, etc. to predict the post-CMP thickness variation at all locations of a chip. Since CMP simulation is a complex task involving several process parameters which may not be available, we also propose alternative analysis and optimization flows that do not rely on CMP simulation, and that moreover consider the sensitivity of linewidth variations to defocus. In our experiments below, we assume that a full-chip topography map is given as input.

Table II.1 shows the change in linewidth of devices in a 2-input NOR gate in the $90nm$ technology when: (1) defocus is changed from $0nm$ to $100nm$, and (2) layout environment (referred to as context) surrounding the gate is changed from isolated to dense. The layout of a 2-input NOR gate is shown in Figure II.3. Isolated context implies that there are no layout features surrounding the cell under study. This simulates the absence of optical proximity effects from neighboring layout features. Dense context implies that the cell is surrounded by other layout features. This simulates the significant optical proximity effects that can result in focus-induced linewidth variation. In our experiment, we place four copies of the

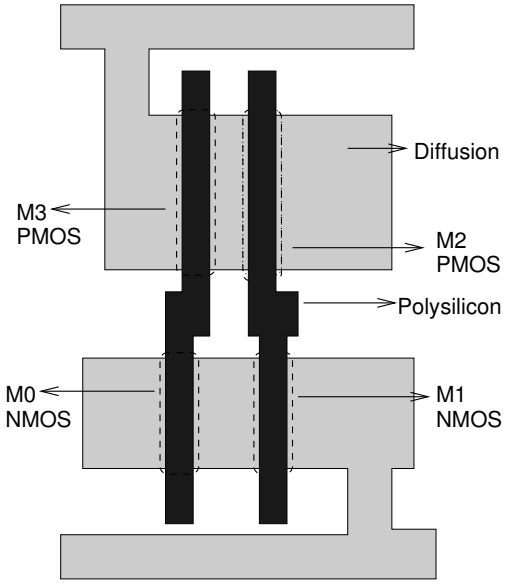


Figure II.3: Layout of a 2-input NOR gate in $90nm$ technology with polysilicon and diffusion layers only. Devices M0, M1, M2 and M3 are labeled on the layout.

same cell on all four sides to simulate a dense context.

Columns 2 and 3 in Table II.1 show the change in linewidth for all devices in a NOR2X2 cell, arising from defocus change (from $0nm$ to $100nm$) for both isolated and dense contexts. Columns 4 and 5 show the linewidth change for all devices due to change in context (from isolated to dense) for nominal ($0nm$) and $100nm$ defocus conditions. From the table we observe that the impact of defocus and pitch on the linewidths of devices in a cell is a large fraction of the total linewidth variation budget, which is typically about 10% of the drawn linewidth.

II.A.2 Systematic Aberration-Induced Variation

Traditionally, variations induced by lens aberration have been considered random due to their small extent. However, as process margins reduce, and as improvements in RETs control variations due to other sources with increased efficacy, lens aberration-induced variations gain importance. For example, our experiments indicate that lens aberration result in high single-digit percentage variation in cell

Table II.1: The effect of defocus and pitch (layout context) on the linewidth of devices in a cell, NOR2X2 for a $90nm$ technology. The change in device linewidth with defocus when the cell is in isolated and dense contexts is shown. The change in device linewidth with pitch, at defocus values of $0nm$ and $100nm$ is also shown. The drawn or target linewidth is $100nm$.

Device	Change with defocus (through-focus variation)		Change with pitch (through-pitch variation)	
	Isolated	Dense	$0nm$	$100nm$
M0	6nm	5nm	0nm	1nm
M1	5nm	3.5nm	0.5nm	2nm
M2	2nm	3nm	1.5nm	0.5nm
M3	1nm	3nm	1nm	1nm

delay for many cells.

Aberration can be described as the departure from ideal imaging induced by an imperfect lens system. Aberration causes optical path differences among the rays, resulting in wavefront deviation from a reference sphere at the exit pupil; this induces blur and distortion of images. Undesirable imaging artifacts from aberration are uncorrectable and, indeed, are sometimes exacerbated through use of RETs such as PSM and OAI [34]. The effects of lens aberration on lithographic imaging [66, 175] include shifts in the image position, image asymmetry, reduction of the process window, and the appearance of undesirable imaging artifacts.

Aberration-induced variations are systematic and depend on location in the lens field. Because proximity effects are well-controlled by RETs, lens aberration is a major source of residual errors in across-field linewidth variation (AFLV) [63]. *Zernike coefficients* capture the deviation from ideal imaging and may be used during lithography simulation to predict the impact of lens aberration on linewidth.

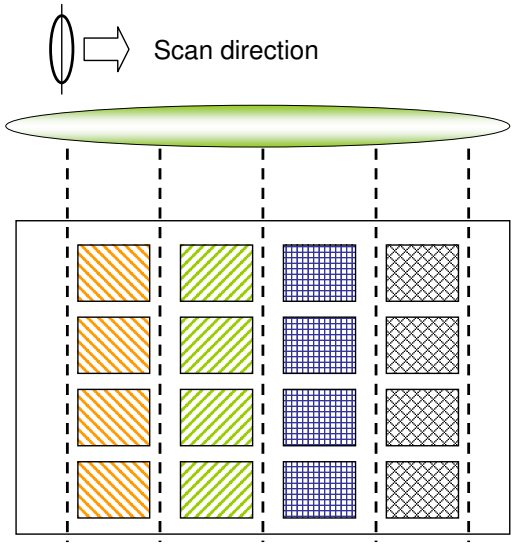


Figure II.4: Linewidth variations induced by lens aberration for different chips in a lens field.

Linewidth Impact From Lens Aberration

Lens aberration parameters (Zernike coefficients), which capture the divergence from ideal behavior of light, change as the slit translates horizontally. *Hence, the linewidth error induced by lens aberration varies along the horizontal direction but stays constant along the vertical direction.* While the variation in linewidth along the horizontal direction is continuous, it is reasonable to discretize it and assume it to remain constant over small regions as shown in Figure II.4.

Using industry-supplied Zernike coefficients at multiple locations in the lens field, we run a lithography simulation on some frequently-used standard cells from a $90nm$ foundry library, and study the impact on linewidth. Figure II.5 shows the average linewidth (i.e., CD) variation of devices in BUFX4, INVX2, NAND2X4 and NOR2X1 cell instances as their position within the lens field is varied. For example, the average gate linewidth variation of NAND2X4 at $100nm$ worst defocus is up to 8nm across the entire lens field. In addition, we investigate the *linewidth skew* (maximum difference in linewidth over all devices in a cell) of different cells. Large linewidth skew unbalances the delays of different timing arcs

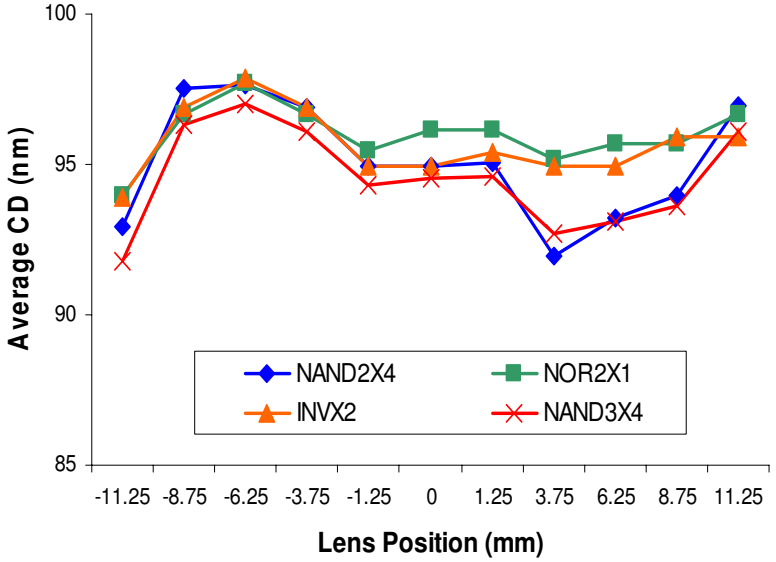


Figure II.5: Average linewidth varies across the lens field; the range of this variation for the NAND2X4 cell is 8nm.

of a cell. Figure II.6 shows the linewidth (CD) skew for NAND2X4 as its position in the lens field is changed. It is evident from these studies that the aberration impact on linewidth error is large across the lens field, and must be modeled to reduce guardbanding and overdesign.

Delay Impact From Lens Aberration

Variations in linewidth directly and indirectly affect circuit delay. At the device level, increase in linewidth causes an approximately linear decrease in saturation on-current of the device which partially determines its delay. Since lens aberration affects different devices in a cell differently, each of the cell's timing arcs can be affected differently. Most standard cells are designed such that the maximum difference in delays of timing arcs (*delay skew*) is small. Due to lens aberration, however, this delay skew can increase, i.e., arcs that are governed by larger than nominal linewidths will be slowed down, while those governed by smaller than nominal linewidths will be sped up.

Figure II.7 shows how the delay, averaged over all timing arcs, changes for four cell masters as the cell instance location is varied from the lens center.

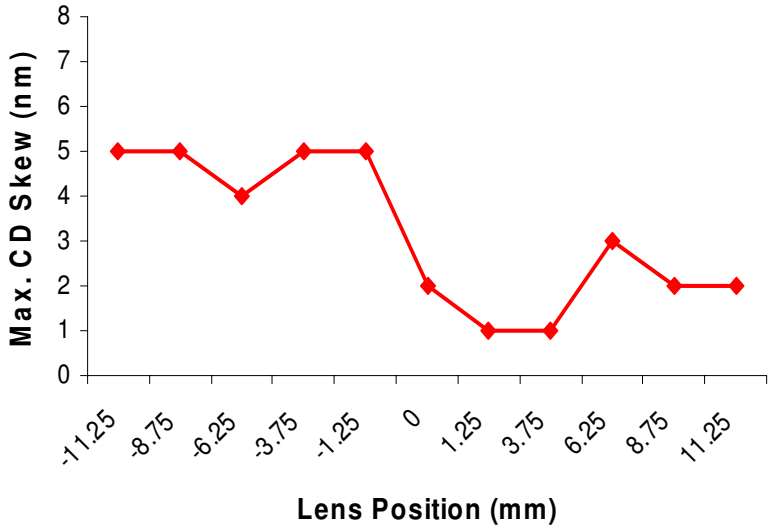


Figure II.6: Maximum linewidth skew among all gates in NAND2X4 cell.

Figure II.8 shows the aberration-induced increase in delay skew with respect to the delay skew of the nominal (or drawn) cell as the location of cell NAND2X4 is varied in the field. The increase is always over 40% because for computation of nominal delay skew, library characterization applies an equal linewidth error to all devices at worst-case process conditions. To compute aberration-induced delay skew, however, lithography simulation is performed at the worst-case process corner and all devices get different linewidth errors.

Linewidth variations also cause variations in cell input capacitance and output slews (transition times). Input capacitance affects the loading of fanin cells and consequently their delays; interconnect delays are also affected. Similarly, slews affect the output slews and delays of cells in the fanout cone. Again, to avoid unnecessary guardbanding, the performance analysis flow (library model characterization, timing/noise analysis, etc.) must comprehend these systematic variations.

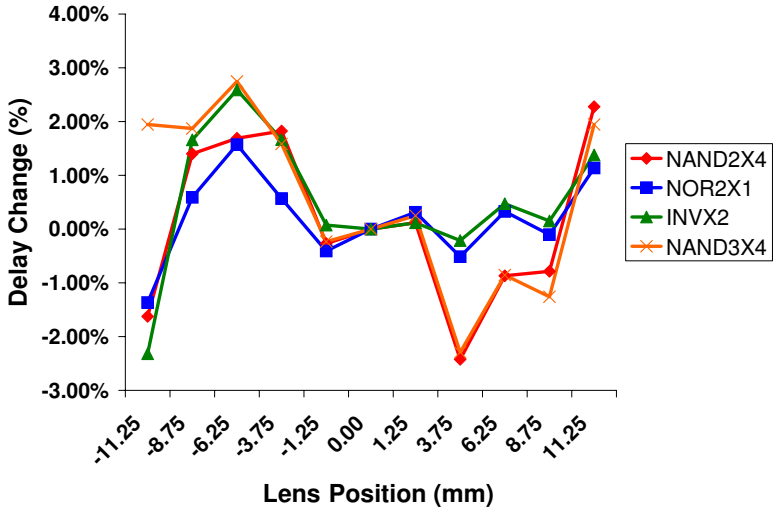


Figure II.7: Change in average arc delay with lens position with respect to center of the lens.

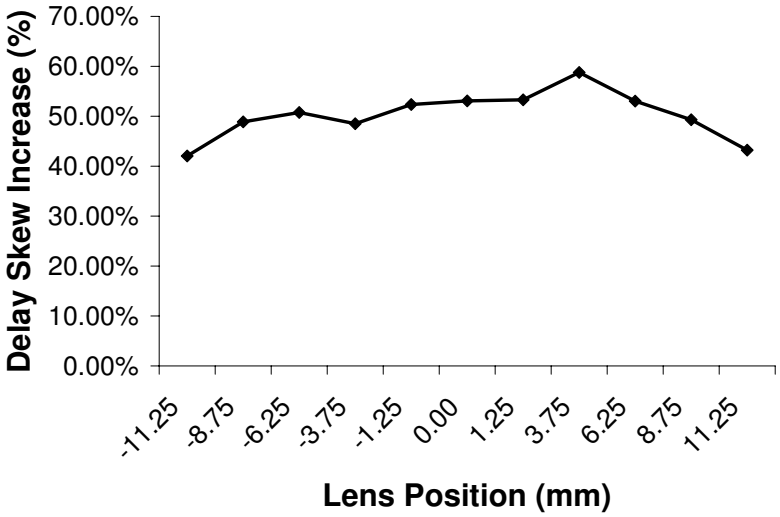


Figure II.8: Percentage increase in delay skew (maximum difference in delays of all timing arcs) of the NAND2X4 cell as lens position is changed, relative to the maximum delay skew of nominal (or drawn) cell.

II.B Defocus-Aware Leakage Estimation and Control

Leakage power is one of the most critical design concerns in sub-100nm technology nodes. Decreased supply voltage (and consequently threshold voltage) combined with aggressive clock gating reduces dynamic power but increases leakage power, causing the leakage share of total power to increase. Leakage is composed of three major components: (1) subthreshold leakage, (2) gate leakage, and (3) reverse biased drain-substrate and source-substrate junction band-to-band-tunneling leakage [20]. In recent technologies, gate leakage has increased dramatically due to gate oxide scaling. However, even at room temperature, subthreshold leakage is the dominant contributor to total leakage at the 90nm technology. Table II.2 compares the subthreshold and gate leakage components for TSMC's 90nm general purpose technology. At the 65nm node, particularly at operating temperatures, subthreshold leakage is expected to again be the dominant contributor [20], and at the 45nm node, the use of high-k dielectrics is expected to significantly reduce gate leakage. With the use of high-k dielectrics, Intel has projected a reduction of 100× in gate leakage at 45nm [43]. Thus, subthreshold leakage is likely to remain the dominant contributor to total leakage for foreseeable technologies. In the remainder of this section we focus on subthreshold leakage, and refer to it as leakage.

Runtime leakage reduction techniques explore design tradeoffs within performance constraints by identifying candidate devices for optimization using leakage power estimates. Inaccurate estimation of leakage power can degrade the results of leakage reduction, underscoring the importance of accurate leakage estimation. Leakage power increases exponentially with decreases in linewidth (also known as channel length or gate length). For example, with 90nm BPTM device models [1, 37], we observe over 5× and 2.5× increase in leakage for PMOS and NMOS devices, respectively, when the drawn linewidth reduces from 100nm to 90nm. In addition to leakage power, manufacturers face the challenge of leakage

Table II.2: Subthreshold and gate leakage of TSMC $90nm$ general purpose nominal V_{th} PMOS and NMOS devices of $1\mu m$ width at two temperatures. Subthreshold leakage is greater than gate leakage.

Device	Temp (^oC)	Leakage (nW)	
		Subthreshold	Gate
PMOS	25	6.45	2.01
NMOS	25	12.68	6.24
PMOS	125	116.80	2.17
NMOS	125	115.90	6.62

variability. Data from [29] indicate that leakage of microprocessors from a single $180nm$ wafer can vary by as much as $20\times$ for a 30% spread in performance. Due to the exponential dependence of leakage power on linewidth, small variations in linewidth can result in significant variations in leakage power.

Traditional leakage optimization techniques are either oblivious to ACLV or model it as a random variable. This results in very pessimistic guardbanding, and hence overdesign. In reality, ACLV due to process variation sources such as focus, exposure, lens aberration and mask errors is partially systematic, and can be modeled. All sources of variations that occur during lithography can be mapped into effective focus and exposure dose variations for the purpose of analyzing their impact on linewidth variation [63].

We exploit the systematic variations in ACLV induced by *focus* variations to estimate and optimize chip leakage power. A similar methodology can potentially be developed to exploit the systematic variations induced by exposure dose variations. We first assess the improvements in leakage estimation that can be obtained by modeling systematic variations in linewidth. In the context of standard cell-based designs, we model linewidth variation in a placement context by simulating the aerial image transfer process during lithography after OPC. To predict leakage of a design, we first analyze its standard-cell layout and extract poly pitch

information. We then use the linewidth model constructed by simulating poly line patterns along with the defocus map of the design to predict post-lithography linewidths. This method does not require design-level lithography simulation to compute post-lithography linewidths. The predicted linewidths are then used to determine device leakages and the circuit leakage.

Our next contribution is to add defocus awareness to enhance a leakage reduction technique, linewidth biasing, that we have previously proposed in [77]. Linewidth biasing increases the linewidth of selected devices (to make the devices slower but less leaky) in cells that are not on timing-critical paths. Defocus awareness enables us to positively bias any cell instances for which devices are likely to print with a smaller linewidth and be extremely leaky. With our modifications, linewidth biasing achieves improved leakage reduction. In summary, the main contributions of our work are:

1. modeling layout- and defocus-dependent systematic components of linewidth variation to better predict leakage, and

2. defocus-aware linewidth biasing that models systematic linewidth variation for improved leakage reduction.

Previous variation-aware leakage analysis methods have focused on statistical analysis (e.g., [149, 40]). In comparison to traditional corner case-based methods, statistical approaches yield a more accurate and less pessimistic analysis. The statistical approaches propose mathematical frameworks within which leakage distributions can be found given the distributions of process variations and the dependence of leakage on them. These approaches assume process variation distributions to be given. Linewidth is assumed to be one of the random variables, and systematic variations in linewidth are modeled using spatial correlations. Such frameworks cannot satisfactorily capture ACLV, which is highly pattern context-dependent. In the absence of suitable statistical frameworks, and for simplicity and easier adoptability, we perform our analysis deterministically.

II.B.1 Defocus-Aware Leakage Estimation

Our defocus-aware leakage estimation methodology is comprised of two modules: (1) linewidth prediction, and (2) leakage calculation. Figure II.9 illustrates the methodology. The linewidth prediction module uses placement information of the design, along with locations of devices within each cell in the cell library (from the cell GDS's), to use the systematic variation in ACLV to predict linewidths of all devices in the design. The leakage calculation module computes leakage of all devices given their linewidths and finds the leakage of the design. We propose the following two flows depending on the availability of die topography information.

- *Defocus-aware, topography-oblivious flow.* We do not rely on a CMP simulator and assume the defocus (due to topography and other sources) to be random. In this flow, we use the fact that linewidth variation is greater for devices with dense or sparse pitches. Devices that have medium pitches, or high pitch on one side and sparse pitch on the other, are self-compensating and print with less linewidth variation [72].

- *Defocus-aware, topography-aware flow.* We consider the availability of a topography map from a CMP simulator. Since topography is a significant contributor to defocus variation, improved topography prediction leads to improved defocus prediction and consequently better leakage estimation.

Defocus-Aware Linewidth Prediction

Both of our flows analyze the layout context of each device of the design and use it with the defocus (assumed to be completely or partly random depending on the flow) at that device location to predict its linewidth. Since leakage is only affected by the dimensions of the gate in MOS devices, we are only interested in linewidth prediction of the gate regions (i.e., overlap of polysilicon and diffusion). Gate regions are nearly always rectangular and are always spaced by the minimum

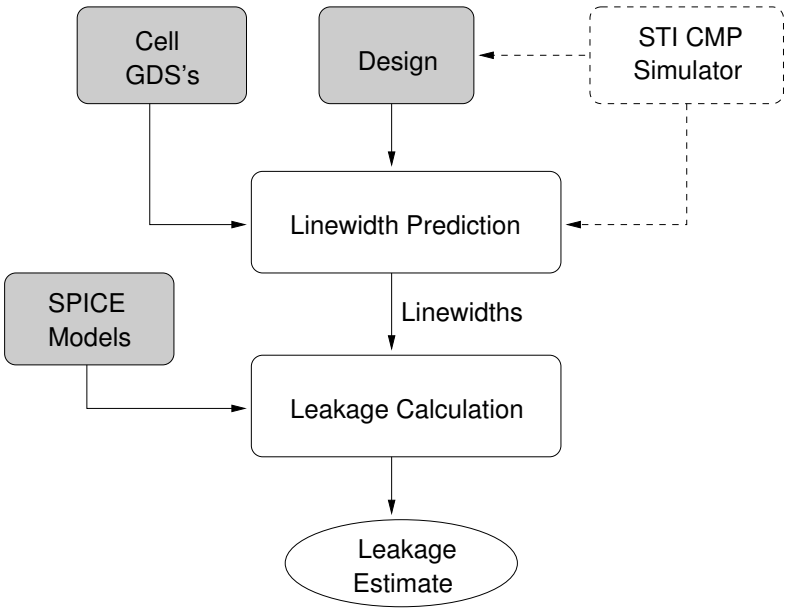


Figure II.9: Our defocus-aware leakage estimation methodology.

design rules from complex shapes such as line-ends and poly bends. Therefore, we can expect a linewidth prediction method that is significantly simpler and faster than lithography simulation to be reasonably accurate.

While in this work we only focus on linewidth for leakage estimation, gate width prediction can further improve leakage estimation. Effects such as diffusion rounding and overlay errors due to misalignment induce error in gate width and can be modeled. As CD variation decreases due to the use of restricted design rules [118], these effects will gain importance.

The main components of linewidth prediction are (1) Bossung lookup table (LUT) generation, and (2) layout analysis for pitch calculation for each device. Figure II.10 illustrates the linewidth prediction methodology. Bossung LUT creation performs lithography simulation to capture and tabulate the linewidth variation with pitch and defocus. Layout analysis calculates the pitch of each device in the design by analyzing the placement and standard-cell layouts.

Bossung LUT Creation

The Bossung LUT captures systematic variations in linewidth due to

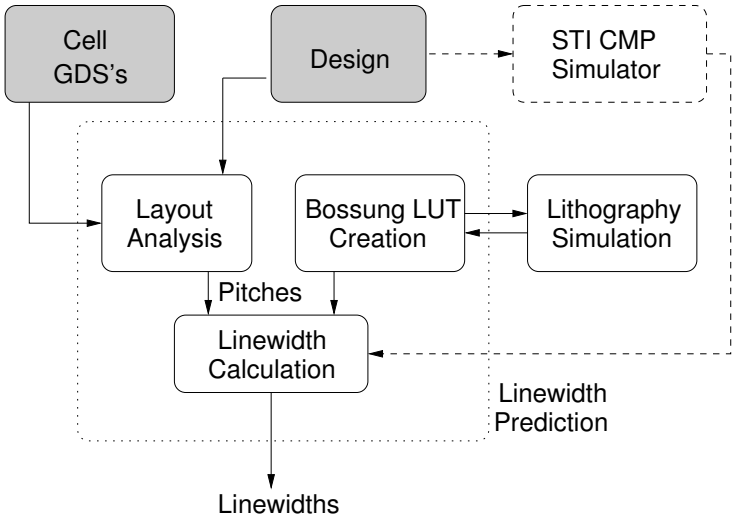


Figure II.10: The proposed linewidth prediction flow.

pitch and defocus. Bossung LUT creation is an offline process that needs to be performed only once for a given cell library and process technology. To create the LUT, we construct line-and-space patterns of gate poly with different spacings to simulate different pitches. The linewidth of gate poly in each pattern is fixed at $100nm$, which corresponds to the linewidth of TSMC $90nm$ technology. Line-to-line spacing is varied from $150nm$ (the minimum spacing at this technology node) to $750nm$ in steps of $100nm$ on both sides. In each pattern, there is one gate poly feature that we call the *poly of interest* with two identical neighbors on each side at various spacings, to get a total of five features in each pattern. Next, for each pattern, neighbors that are away from the poly of interest by more than $800nm$ are removed. It is safe to discard distant neighbors because the $193nm$ steppers used for patterning features in the $90nm$ technology node have an optical radius of approximately $600nm$ (i.e., features separated by more than $600nm$ have negligible impact on each other). We conservatively use $800nm$ as the optical radius for all our experiments. We utilize symmetry of patterns to significantly cut down their number to a total of 153.

After the creation of line-and-space patterns, we perform OPC of the patterns with zero defocus using *Mentor Calibre v9.3_5.9* [6]. To measure linewidth

variation due to defocus, we then perform lithography simulation at different defocus levels for all the patterns. We choose defocus values in the range of ($-200nm$, $200nm$) in steps of $20nm$. Poly linewidth values are then extracted from all simulated printed images at each defocus level. In order to perform the OPC and lithography simulation, we construct a model describing the optical characteristics of wafer stepper and resist coating on wafer. The optical and resist model files are input to OPC and the litho-simulator (e.g., Calibre OPC). The optical model files are generated by specifying the numerical aperture, partial coherence factor, defocus and illumination settings. For our current experimental setup, we generate optical model files for each defocus level with a numerical aperture (NA) of 0.7 in Calibre WorkBench and set the resist threshold to 0.38; both values fall within their standard ranges for $90nm$ OPC setup.

Our Bossung LUT contains rows corresponding to patterns, and columns corresponding to defocus values. Entries in the table give printed linewidth values for the feature of interest in the pattern. For dense patterns, we observe the linewidth to increase by up to $2nm$. For sparse or isolated patterns, on the other hand, we observe a reduction in linewidth of up to $6nm$. These observations are in line with previously reported trends [70]. Further details of our Bossung LUT construction methodology are presented in [92].

Layout Analysis

Given defocus and pitch for a device, the Bossung LUT can be used to predict its printed linewidth. While the defocus is assumed to be completely or partially random, depending on the flow and as described in Section II.B.2, pitch is computed by layout analysis. Pitch of a device is composed of two distances: (1) spacing between its right edge and the left edge of the nearest device to its right, and (2) spacing between its left edge and the right edge of the nearest device to its left.

Figure II.11 illustrates pitch calculation for two devices (A0 and B0) of three neighboring cells with inter-cell and device-to-boundary distances shown. Spacing between devices of a cell can be easily computed by taking the differ-

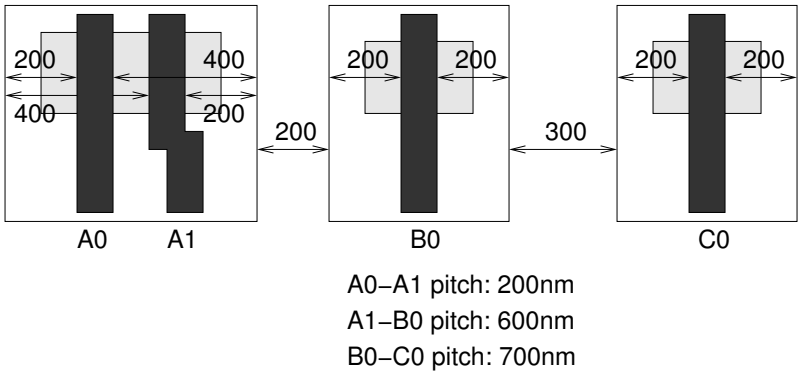


Figure II.11: Pitch computation from a design layout. Nominal linewidth of features is $100nm$.

ence between their respective device-to-boundary spacings for a given boundary. We note that spacings between devices that belong to the same cell need to be computed only once for each standard-cell master in the cell library. Spacing computation between devices of different cells involves adding the inter-cell distance between the two cells and the distance of the two devices from their corresponding cell boundaries, with careful consideration of the cell orientations. Device-to-boundary spacings are available from cell GDS's after performing layout-versus-schematic (LVS) to annotate device gate-poly shapes with device names. Information about neighboring cells, boundary-to-boundary spacing, and cell orientation can be found from the placement of the design (e.g., placed DEF format).

Defocus-Aware Leakage Calculation

We have adapted the methodology proposed by [151] to compute cell gate leakage to calculate cell subthreshold leakage. Subthreshold leakage in a PMOS (NMOS) device occurs only when the gate terminal is in the high (low) state and the source and drain terminals are in opposite states. For each state applied to the inputs of a cell, we propagate the states to all internal terminals of the cell and find the leaky devices. To calculate the leakage of the leaky devices we use a lookup table, characterized with SPICE simulations, that gives the leakage of NMOS and PMOS devices for different linewidths (gate lengths) that we are likely

to encounter. We then sum the leakage of all leaky devices to find the cell leakage for the state. To calculate the average cell leakage, we average the cell leakage over all states; if state probabilities are available, an average weighted by the state probabilities improves accuracy.

In our cell leakage estimation methodology, we ignore the leakage of stacked devices, which is orders of magnitude less than that of non-stacked devices due to self reverse-biasing of stacked devices [88]. Narrow-width effects can be accounted for by characterizing the leakage lookup table for multiple device widths along with multiple linewidths. To compute the design leakage, we sum up leakages of all cells. With respect to SPICE, our approach has a cell leakage estimation error of less than 5% for all cells in our library. Rao et al. [151] also reported similar maximum estimation error for gate leakage.

II.B.2 Experimental Study

We now assess the improvement in circuit leakage estimation from our flow with respect to the traditional corner-based flow.

Experimental Setup

We perform our experiments on the following circuits: c5315 (2,077 cells), c6288 (4,776 cells), and c7752 (3,155 cells) from the ISCAS'85 test suite, and alu128 (11,724 cells) from opencores.org. The circuits were synthesized using *Synopsys DesignCompiler v2003.06-SP1* [11] using a small standard-cell library of 20 cells under tight delay constraints. Our library is composed of the 20 most frequently used cells in our test cases.[1] To create the Bossung LUT, we use *Mentor Calibre v9.3_5.9* [6] for OPC and lithography simulation. Our industry-strength OPC and lithography simulation recipes are for $100nm$ linewidths using $193nm$ steppers. We insert scattering bars (assist features) to improve the process window. We use *Synopsys HSPICE vU2003.09* [13] for all our SPICE simulations and *Ca-*

[1]To identify the most frequently used cells, we first synthesize our testcases using the entire TSMC $90nm$ standard-cell library.

dence SignalStorm v4.1 [4] for library characterization with BPTM BSIM3 SPICE models [1, 37]. Temperature and voltage are assumed to be 25^oC and $1.2V$ in all experiments. We place the designs with *Cadence SOC Encounter v3.2* [5].

We compare (1) traditional, (2) the proposed defocus-aware, topography-oblivious, and (3) the proposed defocus-aware, topography-aware leakage estimation flows. Traditional leakage estimation is corner-based and assumes devices to have the smallest, nominal, and largest linewidths for the worst, nominal, and best cases respectively. The flow involves library characterization with a tool such as Cadence SignalStorm to calculate the leakage of all cells in the library with SPICE simulations. Then, a gate-level leakage analysis tool, such as Synopsys PrimeTimePX [15], sums the leakage of all cells in the design to calculate the design leakage. In the comparisons of the three flows we consistently assume the smallest, nominal, and largest linewidths to be $86nm$, $100nm$, and $110nm$, respectively.

In defocus-aware, topography-oblivious leakage estimation, we assume defocus to be random with a Gaussian distribution ($\mu = 0nm$, $\sigma = 66nm$) leading to a 3σ value of $200nm$. Flagello et al.[63] use 3σ defocus of $300nm$ for their study and hence we consider $\pm 200nm$ defocus to be reasonable. The focus variation assumed for our experimental setup changes between processes and can improve as the process matures. Since variations cannot be completely mitigated, the proposed methodology can be used across any range of focus settings. The assumed defocus of $\pm 200nm$ induces a linewidth variation between $-6nm$ and $+2nm$. Since linewidth variation is caused by factors other than defocus, such as mask errors and exposure variations, we assume a random variation of $\pm 8nm$ in linewidth from other sources. Thus, the contribution of linewidth variation due to defocus is 1/3 of the total linewidth variation.[2] Our assumptions are in line with the findings of [63].

For the defocus-aware, topography-aware flow, we assume the topography shown in Figure II.12 as an input. The topography height is $100nm$ at the center

[2]It is not appropriate to find the standard deviation in linewidth due to the two sources by the "square root of sum of squares" method because contribution due to defocus is partly modeled by our approach, and the remainder is not close to Gaussian. In our setup total linewidth variation is $24nm$ and 1/3 of it (i.e., $8nm$) is assumed to be arising due to defocus.

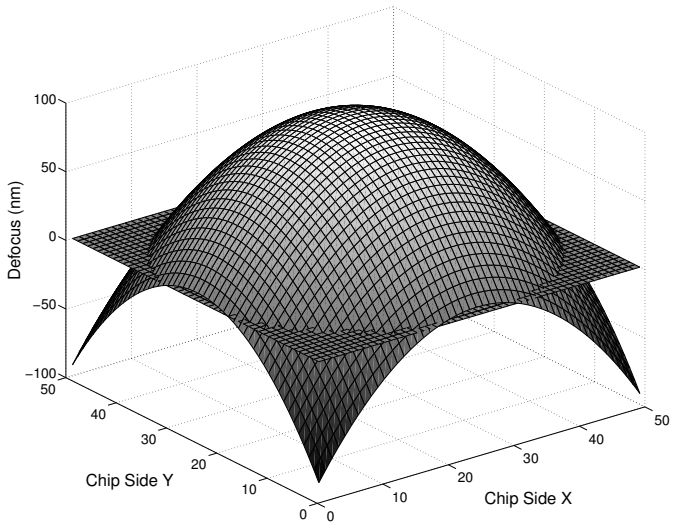


Figure II.12: Die topography used in our experiments. Maximum height is $100nm$ higher than nominal at the center and decreases quadratically with distance from the center to become $100nm$ below nominal at the die corners.

of the die and quadratically reduces with distance from the center to become $-100nm$ at the die corners.[3] A topography variation of $\pm 100nm$ is within the defocus tolerance and expected to exist at the $65nm$ node [69]. At the $45nm$ node, the defocus tolerance is expected to reduce to $\pm 40nm$. We also note that certain steppers also have the ability to compensate for topography variations. In practice, the topography should be predicted by a CMP simulator that models STI layer planarization (STI-CMP simulator) such as those developed in [191, 108]. We again assume the defocus to be $\pm 200nm$, but consider only half of the defocus ($\pm 100nm$) to be random, with the other half being determined from the input topography which alters the defocus by up to $\pm 100nm$.

Results

Table II.3 shows the leakage estimation for all three leakage estimation flows. We observe that the leakage spread between best and worst process corners is

[3]The topography used in Figure II.12 is not unrealistic. Designs that have low device density in the center and high device density towards edges can result in the shown topography.

the largest for the traditional leakage estimation flow. The two defocus-aware flows reduce the spread by decreasing the worst-case leakage and increasing the best-case leakage. The defocus-aware, topography-oblivious flow reduces the spread despite assuming defocus to be completely random because it computes and uses the maximum possible through-focus linewidth variation for each device, which is smaller than the maximum possible linewidth variation experienced by all devices. The defocus-aware, topography-aware flow utilizes the additionally available defocus information to further reduce the leakage spread.

Leakage at the best-case process corner is expected to be the highest for the defocus-aware, topography-aware flow. However, we note that the leakage values for the best-case process corner are identical for the two defocus-aware flows. This happens because leakage decreases nearly exponentially with linewidth and at large linewidths (used for best-case corner), changes in linewidth cause much smaller leakage changes. The difference between linewidths estimated by the two defocus-aware flows is not sufficiently large to register any significant leakage difference at the large linewidths used in the best-case corner. The nominal leakage for the three flows is not directly comparable as it depends on the process and assumed topography.

In addition to accurate design leakage estimation, our methodology predicts the individual cell (or device) leakages for each cell (or device) more accurately. Figure II.13 shows the distribution of the difference between cell leakage predicted by the defocus-aware, topography-aware flow with respect to the traditional method for testcase *c6288* for the best-case, nominal, and worst-case corners. We define the leakage estimation error as the difference between our leakage estimation and traditional estimation. While we observe large cell leakage estimation errors in the range of -29% to 124% for the nominal corner, the error in overall *circuit* leakage estimation is only -1.86%. Our improved cell leakage prediction can be used to improve the quality of leakage reduction techniques that selectively optimize the cells (or devices) with high leakage, such as input-vector control, V_{th} assignment and linewidth biasing.

Table II.3: Estimated leakage power at worst, nominal and best process corners using (1) traditional, (2) topography-oblivious, defocus-aware, and (3) topography-aware (assuming the topography of Figure II.12), defocus-aware leakage estimation flows. Leakage values when the entire circuit uses only low V_{th} devices and when it uses only nominal V_{th} devices are shown.

Circuit	V_{th}	Traditional			Defocus-Aware Topography-Oblivious			
		WC (mW)	Nom (mW)	BC (mW)	WC (mW)	Nom (mW)	BC (mW)	Spread Reduction
c5315	Low	8.006	0.956	0.304	5.269	0.853	0.337	35.96%
	Nom	1.481	0.125	0.036	0.931	0.111	0.040	38.34%
c6288	Low	19.540	2.308	0.726	15.298	2.158	0.838	23.14%
	Nom	3.625	0.302	0.086	2.827	0.282	0.101	22.97%
c7552	Low	12.327	1.469	0.465	9.541	1.360	0.533	24.06%
	Nom	2.281	0.192	0.055	1.757	0.177	0.064	23.94%
alu128	Low	48.499	5.771	1.826	27.264	4.985	1.987	45.84%
	Nom	8.978	0.754	0.217	4.574	0.644	0.238	50.51%

Circuit	V_{th}	Traditional			Defocus-Aware Topography-Aware			
		WC (mW)	Nom (mW)	BC (mW)	WC (mW)	Nom (mW)	BC (mW)	Spread Reduction
c5315	Low	8.006	0.956	0.304	4.119	0.889	0.337	50.90%
	Nom	1.481	0.125	0.036	0.675	0.116	0.040	56.06%
c6288	Low	19.540	2.308	0.726	11.256	2.265	0.838	44.63%
	Nom	3.625	0.302	0.086	1.897	0.297	0.101	49.25%
c7552	Low	12.327	1.469	0.465	7.126	1.433	0.533	44.42%
	Nom	2.281	0.192	0.055	1.203	0.188	0.064	48.83%
alu128	Low	48.499	5.771	1.826	22.442	5.153	1.987	56.17%
	Nom	8.978	0.754	0.217	3.577	0.668	0.238	61.89%

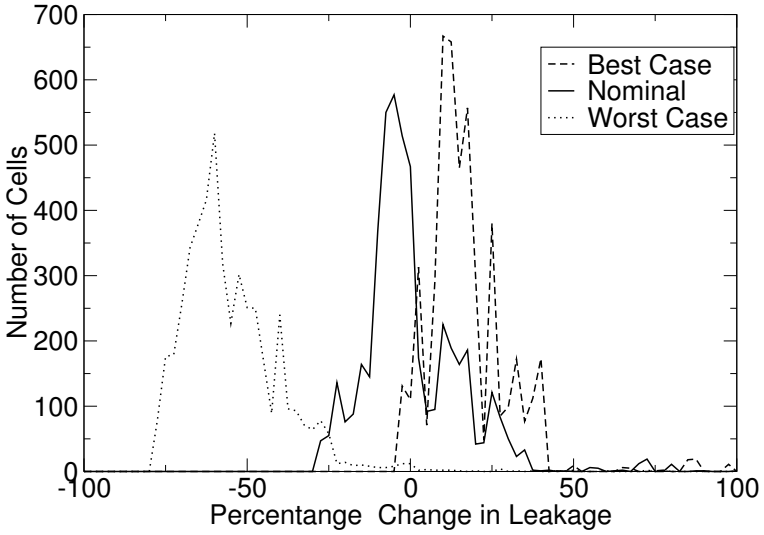


Figure II.13: Distribution of percentage change in leakage estimated with the defocus-aware topography-aware flow with respect to the traditional flow for test-case *c6288* for the three corners. For the nominal corner, the change in total *circuit* leakage is just -1.86% (negative sign implies traditional is higher) but individual cells have larger changes.

II.B.3 Defocus-Aware Linewidth Biasing

Traditional linewidth biasing [77] exploits the fact that leakage reduces exponentially while delay increases only linearly with increase in linewidth. To have minimal impact on circuit delay, the technique selectively biases only the devices that belong to cells that are not on timing-critical paths. Biasing a cell increases its delay and may cause some non-critical paths to become critical, and consequently prevent other cells on the new critical paths from getting biased. Thus, the ordering in which cells are biased affects the quality of leakage optimization. Details of traditional linewidth biasing are presented in Chapter IV but a brief overview is as follows. A *sensitivity*-based greedy optimization is used in which cells are iteratively biased in the order of their decreasing sensitivity. The sensitivity is defined as the ratio of leakage reduction with the slack decrease of a cell caused by biasing. If biasing a cell causes a timing violation, the cell is unbiased (i.e., its linewidth is set back to nominal). The algorithm continues until no more cells can

be biased.

Sensitivity-based algorithms have also been used for V_{th} assignment [161] and gate width sizing [62]. We improve traditional linewidth biasing by more accurately estimating leakage using our defocus-aware estimation flow. This facilitates more accurate sensitivity calculation and better leakage optimization. We use the following terminology to explain our modifications to the sensitivity function for linewidth biasing.

- L_p represents the leakage of cell instance p, L_p^n is the leakage at the nominal process corner, and $\langle L_p \rangle$ is the expected leakage.
- L_{pt} represents the leakage of the t^{th} device of cell instance p, and L_{pt}^n and $\langle L_{pt}^n \rangle$ are its nominal process corner and expected leakages, respectively. $L_p = \sum_i L_{pt}$, where the summation is taken over all devices of the cell.
- ΔL_p^n and $\Delta\langle L_p \rangle$ represent the change in nominal and expected leakages from biasing cell instance p (i.e., biasing all devices in cell instance p).
- Δd_p is the change in delay of cell instance p after biasing it at the nominal process corner.

The sensitivity S_p in traditional linewidth biasing is the ratio between leakage reduction and delay increase of cell p upon biasing, and is given by:

$$S_p = \frac{\Delta L_p^n}{\Delta s_p} \tag{II.1}$$

The sensitivity in defocus-aware leakage estimation is given by:

$$S_p = \frac{\Delta\langle L_p \rangle}{\Delta d_p} \tag{II.2}$$

To compute the expected leakage, we have two flows that are similar to the flows used for defocus-aware leakage estimation, and depend on the availability of topography simulation. The difference is that in our analysis flows we compute the worst-case, nominal, and best-case leakage, while our optimization flows compute the expected leakage as explained below. For the defocus-aware,

topography-aware flow, we assume defocus to be a Gaussian random variable centered at the topography height given as an input from the STI-CMP simulator, and having a 3σ of $100nm$ (50% of our defocus variation budget). For the defocus-aware, topography-oblivious flow, we consider defocus variation to be completely Gaussian random with a mean of $0nm$ and 3σ of $200nm$. We model leakage as a function of linewidth, which in turn is a function of pitch and defocus. Therefore,

$$L_{pt} = \mathcal{L}(\ell(D_{pt}, P_{pt})) \tag{II.3}$$

where D_{pt} and P_{pt} are respectively the defocus and pitch for device t of cell p, and $\ell(D_{pt}, P_{pt})$ represents the linewidth of this device. The expected leakage is

$$\langle L_p \rangle = \sum_t \langle L_{pt} \rangle \tag{II.4}$$

$$\langle L_{pt} \rangle = \sum_t \sum_{D_{pt}} \mathcal{L}(\ell(D_{pt}, P_{pt})).\mathcal{P}(D_{pt}) \tag{II.5}$$

where $\mathcal{P}(D_{pt})$ is the probability that D_{pt} is the defocus value.

Results

A comparison between traditional and defocus-aware (topography-aware) linewidth biasing is presented in Table II.4. While we assume only defocus to be random during optimization (to exploit the systematic dependence of linewidth on defocus and pitch), we present results for the three process corners as described in Section II.B.1. The delay penalty for linewidth biasing is set to 0% (i.e., it is a constraint that circuit delay does not increase after biasing). The runtime penalty due to defocus awareness is under 10% for all our test cases.

Our results show modest leakage reductions for all three process corners from 1.63% to 6.98%. However, given that we have made only minor changes to the sensitivity function of linewidth biasing, we consider these results encouraging. Our approach may be used with several other leakage optimization approaches that

Table II.4: Leakage power after traditional and defocus-aware linewidth biasing. Leakage optimization is performed for nominal process corner and the topography of Figure II.12.

Circuit	Traditional Linewidth Biasing			Defocus-Aware Linewidth Biasing			Leakage Reduction		
	WC (mW)	Nom (mW)	BC (mW)	WC (mW)	Nom (mW)	BC (mW)	WC (%)	Nom (%)	BC (%)
c5315	3.948	0.855	0.326	3.838	0.838	0.321	2.78	2.01	1.63
c6288	9.363	1.923	0.730	8.958	1.861	0.712	4.33	3.23	2.56
c7552	6.678	1.350	0.507	6.212	1.280	0.485	6.98	5.17	4.21
alu128	21.258	4.908	1.907	19.968	4.663	1.827	6.07	4.99	4.19

rely on identifying candidate cells or devices to make tradeoffs. Larger leakage reductions are expected when the impact of systematic linewidth variations on gate delays is also considered during optimization. Slacks, created when pessimism in delays is reduced by systematic variation-aware timing analysis, can be used towards leakage reduction. The extent of leakage reduction depends on the reduction in pessimism and the effectiveness of the leakage reduction knob to tradeoff delay versus leakage.

II.C Detailed Placement for Leakage Reduction Using Systematic Through-Pitch Variation

As discussed above, ACLV has a substantial systematic component [145] which is predictable once the *pitch* and *defocus* of a device (line) are known [63, 70]. Pitch of a device captures the context of the gate of the device and in simple terms is the spacing of the gate from neighboring gates. Once placement has been performed, pitches of all devices in the design are known.

Through-pitch variation is the linewidth variation that occurs over different permissible pitches. RETs such as OPC and scattering bar insertion reduce but do not completely eliminate through-pitch variation, especially at non-ideal defocus and exposure conditions. For the 65nm technology that we study, the through-pitch variation, after RET application, is 5nm at zero defocus (ideal focus condition) and 12nm at the maximum defocus value of 100nm. Such variations in linewidth translate to 100% and 527% variations in NMOS device leakage, respectively. Fortunately, most devices have pitches that are less sensitive to defocus, and the expected defocus value is smaller than the maximum.

In this section, we propose a novel detailed placement technique that changes the placement of cells to change the pitches of devices and consequently reduce their leakage. Cells can be composed of several devices; pitches of the devices that are closest to the cell boundaries (henceforth referred to as *boundary devices*) change with placement. Placement has negligible impact on the pitches of devices other than the boundary devices; this is due to their large distance from the boundary and the fact of their being shielded by the boundary devices of the cell. However, most commonly used cells such as small- to moderately-sized buffers, inverters, NANDs, and NORs have all of their devices near boundaries. For such cells, device pitches and consequently leakage will be affected by detailed placement. For example, the leakage of a NAND gate (NAND2X1) changes by 18% when it is sandwiched between two other NAND2X1 gates versus when it has no neighbors.

Our methodology involves two steps. First, a matrix is constructed to capture the leakage when two cells are placed next to each other. This matrix is used to drive our optimization and to evaluate the leakage of a given placement. Second, we divide the design into small windows and optimize the windows individually. During the optimization cells are redistributed in rows, and within each row their ordering, spacing, and orientation are optimized using a traveling salesman formulation. We ensure that the timing-critical cells remain unaffected during optimization to minimize the impact on their delays and the delays of their interconnects.

A recent work by Hu et al. [85] proposed a pattern-sensitive placement approach for minimizing linewidth variation. The work by [85] was published in the period between the submission and acceptance of our work [91]. The objective of [85] is to minimize total edge placement error (EPE)[4] by modifying detailed placement subject to wirelength constraints. Although the approach of our work and that presented by Hu et al. appear similar, they differ in several aspects. Hu et al. seek to minimize the overall EPE variation while we seek to to minimize leakage power. Hu et al. do not consider placement of filler cells[5] while we explicitly optimize their placement. Consideration of filler cells in detailed placement is very important, since filler cells in sub-90nm technology have non-functional poly and their placement alters poly pitches and, consequently, device leakage.

In Section II.B, we described a leakage analysis and optimization methodology. That approach calculates the pitches of all cells in the design and finds the susceptibility of the cells to defocus-induced linewidth variations. The cells that are more susceptible are preferred for optimization over the others. The approach in this section modifies the pitches themselves to reduce leakage, and is therefore complementary to the method described in Section II.B. Gupta et al. [74] proposed a placement perturbation approach to increase the number of scattering bars that can be inserted. While [74] can increase the number of scattering bars and reduce through-pitch variations, design objectives such as delay and leakage are not targeted. The approach of [74] is also limited to perturbing cells in the neighboring free space in a single row, which limits the opportunities for optimization. The use of detailed placement to enhance the printability of cells has also attracted interest from the industry recently [68].

II.C.1 Detailed Placement

Traditionally, placement is separated into two phases – global placement and detailed placement. Global placement generates a legalized (i.e., with no over-

[4]EPE is a measure of linewidth variation.

[5]Filler cells are placed in the empty space between actual cells to maintain power and ground rail connectivity.

laps) placement of standard cells in rows. Detailed placement is a refinement step which performs small-range perturbations to generate a new optimized placement. Several approaches to detailed placement have been proposed, with most focusing on wirelength minimization (e.g., [58]) or timing [53]. Our approach, to the best of our knowledge, is the first to consider the impact of detailed placement on poly gate pitch to reduce leakage which is strongly and systematically dependent on pitch.

Placement can change the pitches of boundary devices of a cell by using the following three knobs:

- Neighbor selection. Different cells have different spacings between the boundary and the boundary devices. Thus the neighbor of a cell affects the pitch of the cell's boundary devices.

- Orientation. Within a cell the spacing between the left boundary and the closest boundary devices is different from the spacing between the right boundary and the boundary devices closest to the right boundary. Thus, the orientation of a cell (i.e., "flipped" or not) affects the pitches of the boundary devices.

- Cell-to-cell spacing. In general, introduction of space between two cells causes the pitches of the boundary devices of the cells to become sparse (i.e., large). However, as explained later, the presence of non-functional polys in fillers (which are always inserted into any space between two adjacent cells) decreases the pitches of the boundary devices in the neighboring cells. Cell-to-cell spacing affects the pitches of the boundary devices irrespective of the fillers containing polys.

Figure II.14 shows two placements of three cells in a row and how the pitches of the gates in the cells change.

Fillers are always inserted into any space between two adjacent cells to ensure connectivity of the power and ground rails. In $65nm$ and beyond technologies, fillers may have non-functional polys to enhance layout uniformity. Such

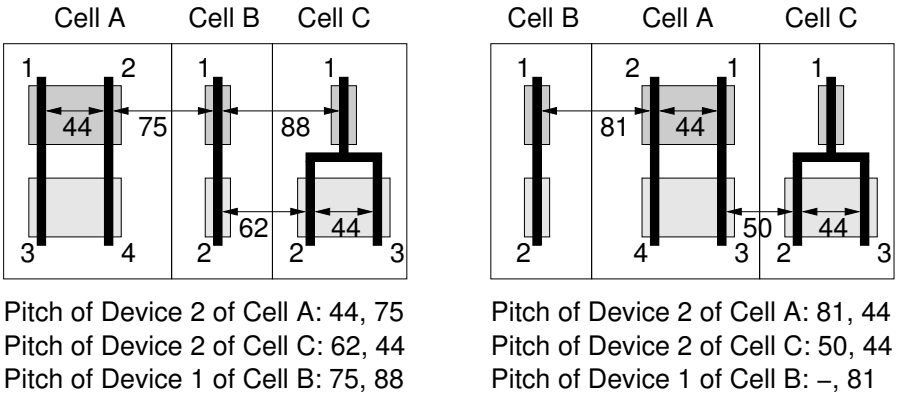


Figure II.14: Detailed placement affects device pitches. Two placements of three cells in a row, and the device pitches, are shown.

fillers decrease the pitch of the devices in neighboring cells (i.e., make the pitch dense). On the other hand, fillers that do not have polys increase the pitches of the devices in neighboring cells (i.e., make the pitch sparse). In both cases filler insertion is a powerful knob to control device pitches and is considered in our approach.

II.C.2 Assessing Leakage Impact of Detailed Placement

Potential leakage savings from detailed placement depend on the following factors.

- Pitch range. The minimum pitch attainable by detailed placement depends on the spacing from boundary device to boundary of the cells. When fillers contain polys, the maximum pitch is attained when two cells with large boundary device to boundary spacing are placed next to each other. If fillers do not have polys, the maximum pitch is attained when fillers are inserted between cells. A larger difference between minimum and maximum attainable pitches affords greater leakage reduction. For our $65nm$ library, the spacing between neighboring gates of any two cells when they abut varies between $210nm$ and $520nm$.

- Linewidth variation due to pitch. This is process-dependent. We expect larger leakage reduction if the linewidth variation due to pitch is large. For our $65nm$ process, linewidth is $60nm$ for {210nm, 210nm} pitch (where first and second distances in the tuple are respectively the left and right spacings to immediate neighbors) and $56nm$ for $\{520nm, 520nm\}$ pitch at $0nm$ defocus. At $100nm$ defocus, the linewidth is $60nm$ and $51nm$ for the two pitches, respectively.

- Leakage reduction with linewidth. Leakage decreases exponentially with linewidth increase. Larger leakage change with linewidth change allows more leakage optimization by detailed placement. For our technology, PMOS and NMOS leakages increase from $0.383\mu A/\mu m$ and $0.270\mu A/\mu m$ to $1.868\mu A/\mu m$ and $0.887\mu A/\mu m$, respectively, when linewidth decreases from $60nm$ to $51nm$.

We construct a $\Delta leakage\ matrix\ L$ to capture the leakage change when a cell is placed next to another cell, with respect to when it is placed without any neighbor. L additionally needs to capture the fact that the leakage change depends on which particular sides of the two cells touch. Thus, the matrix has two rows, and two columns, corresponding to the two sides of each given cell. The matrix is constructed only once for a library; if there are N cells in the library, the matrix has $2N$ rows and columns. In the following, we use *Side 0* and *Side 1* to denote the left and right sides of a cell, respectively. Then,

$$L_{ij} = \Delta leakage_{\lfloor i/2 \rfloor} + \Delta leakage_{\lfloor j/2 \rfloor} \tag{II.6}$$

when Side $i\%2$ of Cell $\lfloor i/2 \rfloor$ touches Side $j\%2$ of Cell $\lfloor j/2 \rfloor$. Here, e.g., $\Delta leakage_{\lfloor i/2 \rfloor}$ is the leakage change of Cell $\lfloor i/2 \rfloor$ with respect to when it has no neighbors.

Leakage calculation when two cells abut consists of two parts.

1. Linewidth calculation. We use the linewidth calculation methodology described in Section II.B. Device pitches can be computed from device to

boundary spacings for all devices in the two cells. Defocus depends on the process conditions and is difficult to predict; so we assume it to be a random variable with normal distribution ($\mu = 0nm$, $\sigma = 33.3nm$). We use the calculated pitch value and the defocus distribution to find the distribution of linewidth. Lithography simulation can alternatively be used to generate a more accurate linewidth distribution, albeit with higher runtime.

2. Cell leakage calculation. We use the leakage calculation methodology described in Section II.B. To calculate the device leakage distribution from the linewidth distribution, we use a leakage lookup table characterized with SPICE for a variety of gate width and gate length (linewidth) values. We then calculate the expected value of device leakage for all devices in the two cells, and use them to calculate cell leakage. Using logic propagation in the cell, we find the fraction of states in which each device leaks and call it the *off-fraction*. Leakage of a cell is the sum of leakages of its devices weighted by their respective off-fractions.

Our matrix construction methodology is fast and practical for large libraries. We note that such a matrix needs to be created for the corner of which leakage analysis and optimization is desired. In our studies, we use the typical-leakage corner which is typical process, $1.1V$, and 85^oC (PVT).

The matrix abstracts the pitch impact on leakage that arises due to through-pitch ACLV for use in optimization. Such a matrix may be created by the process engineers and library designers, and can be used by circuit designers to evaluate and optimize leakage.

II.C.3 Leakage Optimization

Given the impact of placement on leakage, we now present a detailed placement technique that minimizes leakage. Certain cells can be critical to the optimization. For example, low-V_{th} cells are more critical than standard-V_{th} and high-V_{th} because they have larger (absolute) leakage reductions; similarly,

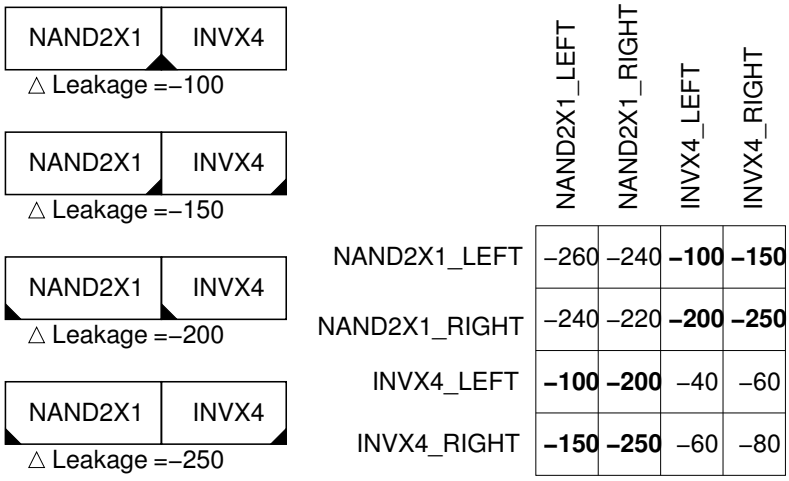


Figure II.15: Creation of Δleakage matrix L. The bold entries are found by placing NAND2X1 and INVX4 next to each other. Non-bold entries are found by placing NAND2X1 next to another NAND2X1, and by placing INVX4 next to another INVX4.

standard-V_{th} cells are more critical than high-V_{th}. Cells that have fewer devices, such as small- to moderately-sized inverters, buffers, NAND's, and NOR's, are the most affected by proximity and are more critical to address in the optimization. Our optimizer maximizes leakage savings for such cells.

We dissect the design into small windows and run the optimization for each window. Such a method is effective for our purposes because of the localized impact of proximity which does not hold, for example, with total wirelength objectives. The optimization relies on having a rich set of boundary-to-device spacings and whitespace to reduce leakage. Even a small window containing 15 cells offers enough scope for optimization. Smaller windows restrict the movement of cells to within smaller boundaries, and hence the wirelength increase is bounded. Moreover, smaller windows are faster to optimize and different windows may be simultaneously and independently optimized on multiple CPUs. Prior to the optimization, all fillers are removed; they are inserted back into the whitespace after optimization.

To simplify the explanation of our optimization, first assume that there

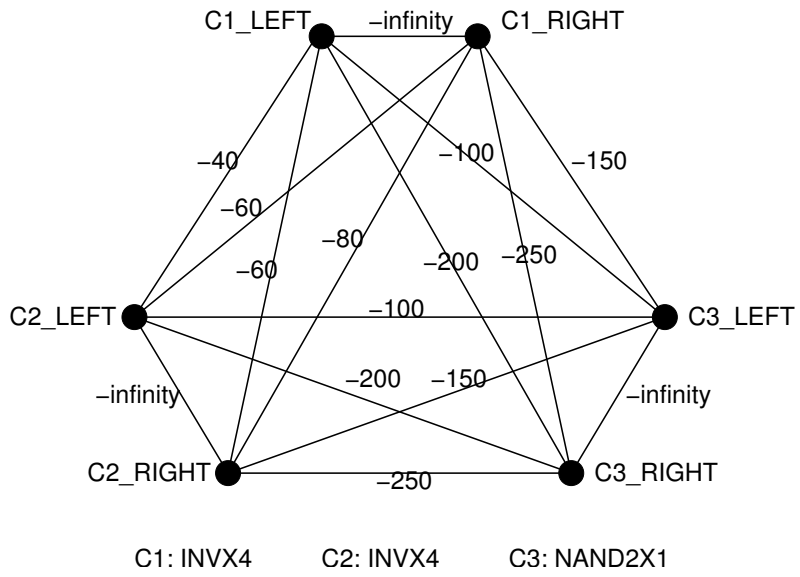


Figure II.16: Creation of the graph for three cells C1, C2, and C3. C1 and C2 are of types INVX4 and C3 is of type NAND2X1. Edge weights are derived from the matrix L shown in Figure II.15. For edges between two vertices that belong to the same cell, a weight of $-\infty$ is assigned.

is only one row in the window, and that there is no space for fillers in the row. Under these assumptions, the problem is to identify an ordering of the cells, along with the cells that must be flipped to yield the maximum leakage reduction (i.e., minimize the sum of $\Delta leakage$ for all cells in the window). We transform this problem to the well-known traveling salesman problem (TSP) [107] as follows.

- We create two vertices for every cell – one for the left side and another for the right side.
- Edge weights or distances between vertices denote the leakage reduction when the sides represented by the vertices touch. These weights are obtained from the Δleakage matrix L.
- Weights of edges between vertices that denote sides of the same cell are set to $-\infty$ since the two vertices must always occur consecutively in the TSP tour.

We can now use a standard symmetric TSP heuristic capable of handling large negative weights to obtain a TSP tour. The order of vertices in the tour gives the order in which cells must be arranged in the row, with their orientations, for maximum leakage reduction. We solve the TSP with the *multifragment greedy* algorithm [25] that considers edges in increasing order of weight for insertion into the tour.

Space Allocation for Fillers

We now lift the assumption that no space is available for filler cells. The optimization must additionally space the cells and insert fillers to minimize leakage. Toward this end, we calculate the leakage reduction of all cells in the library when a filler cell of minimum width (FILLX1) touches an edge of the cell. The matrix L is expanded by one row and one column for the FILLX1 cell. We assume that proximity effects do not range beyond the minimum-width filler, and that the leakage of cell does not depend on the size of the filler. This assumption is valid when fillers contain dummy polys as well as when they do not. When fillers contain polys, the separation between a boundary poly and the boundary is identical for all fillers. This is generally true because fillers other than the smallest filler are essentially abutting copies of the smallest filler. Therefore, the pitch of a neighboring cell is the same, irrespective of the filler size. When fillers do not contain polys, the width of the smallest filler is typically sufficient to keep the devices of the next cell outside the optical radius. Larger fillers push the neighbor device even further away but devices beyond the optical radius have negligible proximity impact so all fillers have nearly identical pitch impact.

Our graph for TSP requires the following changes:

- We add vertices corresponding to FILLX1's; the number of added vertices is equal to the number of FILLX1's insertable in the row.
- The weight of the edges between fillers is set to zero and the weight of the edges between fillers and cells are obtained from the Δleakage matrix L.

For example, if space is available for 20 FILLX1's, we make the following

modifications to our TSP.

- Add 20 new vertices to the graph.
- Set the weights of the edges between them to zero.
- Set the weights of the edges between a vertex representing an side of a cell and all vertices representing fillers from the matrix L to reflect the leakage reduction when the cell's side touches a filler.

As before, we solve a symmetric TSP with the multifragment greedy heuristic. If two fillers are consecutive in the tour, two FILLX1's must be placed next to each other. Two abutting FILLX1's are identical to a FILLX2, so we replace multiple consecutive fillers with larger fillers. We evaluate the quality of our TSP-based single-row placement solution against an optimal solution found by enumerating all possible single-row placement solutions for two arbitrary cells. Table II.5 compares the leakage results normalized against the maximum leakage (which is also found by enumeration). Our approach is consistently able to attain near-optimal solutions with significantly less runtime for other configurations as well.

Multiple Rows

We now eliminate the assumption that there is only one row in the window. We exhaustively partition the set of cells into the rows and optimize these partitions using the single-row optimization. The number of partitions can be computed as a sum of Stirling numbers of the second kind and is nearly exponential in the number of cells in the window. However, a large number of these partitions can be pruned due to row capacity constraints and because of multiple instances of the same cell master (which are alike) in the window. Further, best single-row results for some rows can be cached during the partitioning. With these runtime improvements, our approach handles up to two rows with ease, and handles three rows with feasible runtime (approximately $2s$ per window) assuming ≤ 20 cells in the window.

Table II.5: Leakage comparison of TSP-based placement against optimal found by enumerating all placements. Leakage normalized against maximum leakage. Cell set 1 is {INVX1, INVX1, INVX1, NAND2X1, NAND2X1, AOI22X1, AOI22X1} and cell set 2 is {INVX2, INVX2, NOR2X0, NOR2X0, NOR2X0, MUX2X0, MUX2X0, MUX2X0}

Cell Set,	Max.	Optimal		TSP-based	
#Fillers	Leakage	Leakage	CPU (s)	Leakage	CPU (s)
1, 0	1	0.928	0.22	0.932	0.030
1, 5	1	0.804	270.18	0.806	0.034
2, 0	1	0.976	1.08	0.976	0.033
2, 3	1	0.922	221.09	0.922	0.034

Minimizing Timing Impact

The perturbation of detailed placement from the original placement results in wirelength change, which can impact wire parasitics and consequently timing. Even though our localized placement perturbations do not significantly affect timing, small changes in the timing of critical paths can affect the minimum clock cycle time. To minimize the timing change of critical paths, we fix the cells and nets in the critical paths: fixed cells are not moved during optimization and fixed nets are not changed during the ECO routing that is performed after optimization. Since the nets in the critical path are fixed, all cells connected to these nets should also be marked as fixed and not moved during optimization. (Even despite such measures, the delay of such nets can change marginally due to the coupling capacitance with neighboring nets, the routing for which may change.) We also fix all flip-flops, clock buffers, and clock nets to avoid any impact on the clock tree.

During optimization, for each cell marked as fixed, we break the row in which the cell is placed into two parts: left of the cell and right of the cell. The two parts are optimized individually; this ensures that the fixed cells do not move and that no other cells overlap with the fixed cells. Although we do not move

fixed cells during optimization, our approach considers their location during the placement of other cells. Our overall algorithm is presented in Figure II.17. Given an original placement, list of critical cells, Δleakage matrix L, and a window size, the optimization outputs a final placement with lower leakage.

Minimizing Wirelength Increase

Wirelength increase is undesirable because it can cause congestion and degrade routability. Also, wires act as capacitive elements and longer wires can increase dynamic power. We expect a smaller window size to cause smaller wirelength increase, because the movement of cells during optimization is restricted to within the windows. To reduce wirelength, we run the optimization in phases with each phase successively increasing the window size until a *final window size*, which is a user input, is attained. The result of a phase is only accepted if it improves upon the result of the previous phase by more than a threshold (set to zero in our experiments). This policy has the effect that cells are moved farther only if the leakage reduction is greater than the threshold.

II.C.4 Experimental Study

In this section, we discuss the details of our experimental setup for optimization followed by detailed routing, and present results.

Experimental Setup

A high-level overview of our experimental setup is shown in Figure II.18. To setup the optimization, we first perform synthesis of testcases using multiple threshold voltage libraries (high-V_{th} and normal-V_{th}) in 65nm technology. We then perform placement, and detailed routing, followed by extraction and timing analysis. From the timing analysis result, we identify all timing-critical cells for input to the optimizer. We also create a set of nets corresponding to the critical cells that should not be touched during ECO routing of the optimized detailed placement.

The optimizer reads in the placed and routed design, *fixed cells* list, and

Input: Placed design; timing critical cells, Δleakage matrix L (of Figure 1) that denotes leakage change when any two cells touch; window size

Output: New placement with lower leakage and small/no delay impact

[1] $D \leftarrow$ {critical cells} $\cup$ {cells connected to output nets of critical cells}

[2] **forall** windows w in the design

[2.1] $C \leftarrow$ {All cells in w}

[2.2] PartitionAndPlace($C - D$, w)

PrartitionAndPlace(C, w)

[1] r = firstRowOf(w)

[2] $bestCost = \infty$

[3] **forall** $S \in$ Subsets(C)

[3.1] **if** (rowCapacity(r) < $\sum_{c \in S}$width(c)) // row capacity not exceeded

[3.1.1] <$tour$, $Cost_r$> = TSPPlace(S, r) // Place cells S in row r

[3.1.2] $Cost_{w-r}$ = PartitionAndPlace($C - S$, $w - r$) // Place remaining cells in remaining rows

[3.1.3] **if** (bestCost > $Cost_r + Cost_{w-r}$)

[3.1.3.1] $bestCost = Cost_r + Cost_{w-r}$

[3.1.3.2] $bestTour = tour$

[4] save(r, $bestTour$)

[5] return $bestCost$

TSPPlace(C, r)

[1] F = width(r) - $\sum_{c \in C}$width(c) // Number of fillers

[2] $G = (V, E)$; $V = C \cup_1^F$ {"FILL"}; Construct E from L // Insert cells in C and F fillers into V

[3] Solve TSP on G

[4] return <tour from TSP, cost of tour>

Figure II.17: Detailed placement pseudo-code for leakage optimization.

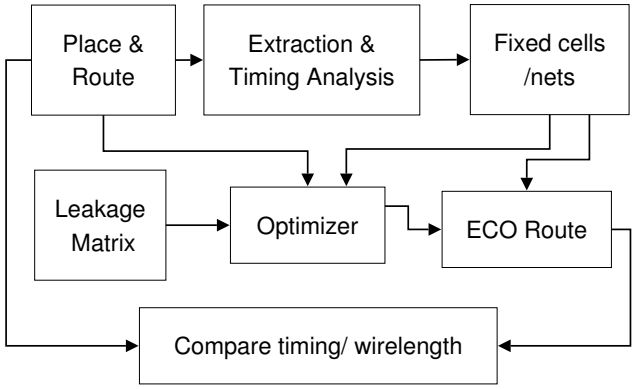


Figure II.18: Our experimental flow.

leakage matrix L, and performs placement optimization. The optimizer outputs a legal layout (in which the cell orientations are correct and no cell overlaps with any other cell). We then perform ECO routing on the optimized result with the constraint that fixed nets are not re-routed, and then perform parasitic extraction and timing analysis.

The most important steps in our experimental setup are leakage matrix construction and interfacing the optimizer to the router. These steps are discussed in greater detail in the following.

Leakage Matrix Construction

Bossung LUT provides linewidth of devices in different layout contexts. To construct the Bossung LUT, we take as input a standard-cell layout in $65nm$ technology and compute poly pitch for all the devices in the layout. This is performed by analysis of neighborhood of each device in the layout. We use a device layout analysis tool (built on the OpenAccess API [7]) to search the neighborhood of every device and compute the spacing to other devices. To obtain the printed linewidths for devices at different pitches, we take as input their litho-simulated device contours (generated *after application of scattering bars and OPC*) at specific defocus values. We interpolate between defocus values using a 2^{nd}-degree polynomial [121]. We use the procedure outlined in the steps above to obtain linewidths of devices in different cell layout contexts for construction of the leakage matrix.

Table II.6: Testcases used in experimental validation.

Circuit	#cells	Max. Speed (MHz)	Leakage (mW)	Dynamic (mW)	Wirelength (mm)
AES (80% util.)	18665	413	0.6211	0.4002	330.87
AES (85% util.)	18726	419	0.6323	0.4127	320.65
DES	79419	417	6.3083	3.6754	1146.14

We assume a normal distribution for defocus with a mean of $0nm$ and σ of $33.3nm$.

Optimizer – ECO Route Interface

To optimize a given detailed placement for minimum leakage, we start from an existing placed and detail-routed design. We perform timing analysis on the design to identify all timing critical paths. We then choose all cells from critical paths that have a slack value ranging from the minimum value to 5% of the clock cycle time and mark them as fixed. Since the nets connected to these fixed cells also cannot move, we create a *dont_touch* list of nets connected to all fixed cells. All route segments corresponding to fixed nets are not moved during ECO routing. To prevent disruptions to the routing of these nets, we update the existing fixed cells list to include all cells connected to fixed nets.

We use *Cadence RTL Compiler (v5.1)* [3] for multi-V_{th} synthesis of our testcases. For our experiments, we use the 50 most-frequently used cells from high-V_{th} and nominal-V_{th} libraries. To run placement, clock tree synthesis, routing and timing analysis, we use *Cadence SOC Encounter (v5.2)* [5]. Dynamic power measurement was performed with *Synopsys DesignCompiler (vY-2006.06-SP5)* [11]. The details of our testcases are shown in Figure II.6. The standard-cell row utilization for our testcases: *aes* and *des* (available from *opencores.org* [8]) are 80% and 73% respectively. To demonstrate leakage reduction at higher utilizations, we implemented *aes* at 85% row utilization. Row utilizations greater than 80% are not common because of routing congestion concerns. We built our optimizer on top of *OpenAccess API (v2.2.4)* [7].

The inputs to our optimizer are a routed design and a list of fixed cells. The output from the optimizer is the design with modified placement with "dangling" wires for some cells (since locations of some cells are perturbed during optimization). We feed the output from the optimizer to SOC Encounter and invoke the router in ECO mode along with *dont_touch net* router directives.[6] After ECO routing, we perform parasitic extraction and timing analysis to evaluate change in wirelength and timing. We use the worst-case corner for timing analysis in our flow.

Results

We evaluate the proposed approach for leakage reduction and change in wirelength, delay, and dynamic power. Table II.7 presents our results for the three testcases and multiple window sizes. The leakage reduction upper bound indicates the maximum leakage savings possible if the lowest-leakage context for all cells could be created by choosing their neighbors. The upper bound may not be attainable because only the cells available in the window can be used as neighbors, and because of limited availability of free space. The reported leakage reduction, wirelength change, and delay change are with respect to the original placed and routed design. Wirelength is the actual routed wire length after detailed routing; in all cases detailed routing finished without any violations.

From the results, window sizes of $6\mu\times$ 2 rows and $4\mu\times3$ rows offer good solution quality with feasible runtime. We observe that the effect of our leakage optimization on maximum frequency is marginal. For the testcases AES and DES, 10.97% and 23.57% of the cells are marked to be fixed during optimization. Without marking these cells as fixed, the maximum frequency drops by 5.62% for AES while the leakage reduction only increases from 6.41% to 7.45%. While the effect of the optimization on wirelength is not negligible, without our wirelength reduction policies the wirelength increase is 12.33% for AES (80% utilization) in comparison to 8.14% with the policies. This shows that our wirelength- and de-

[6]To direct SOC Encounter to honor existing routing of fixed nets, we use the "setAttribute -net $< NET\_NAME >$ -skip_routing true" command prior to invoking ECO route.

Table II.7: Assessment of impact on leakage, wirelength, and delay of the proposed technique. The rows annotated with † symbol correspond to results without the use of delay and wirelength reduction policies. LRU refers to the leakage reduction upper bound computed ignoring the available space contraints.

Circuit	Final Window Size	Leakage Decrease (%)	ΔWire Length (%)	ΔMax. Frequency (%)	ΔDynamic Power (%)	CPU (s)
AES	4μ×1 row	2.91	+0.72	+0.33	+0.24	5.18
80%	6μ×1 row	4.16	+2.39	-0.41	+0.82	8.72
util.	8μ×1 row	5.08	+4.94	-1.18	+1.70	14.64
	4μ×2 rows	5.21	+3.86	+0.50	+1.27	37.90
LRU	6μ×2 rows	6.41	+8.14	-0.49	+2.58	301.35
=8.95%	2μ×3 rows	4.02	+2.08	+0.46	+0.69	23.83
	4μ×3 rows	6.44	+7.12	-0.41	+2.45	1964.09
	6μ×2 rows†	7.45†	+12.33†	-5.62†	+4.08†	284.34†
AES	4μ×1 row	1.81	+0.93	+0.21	+0.38	5.23
85%	6μ×1 row	2.77	+2.65	-0.33	+0.90	9.57
util.	8μ×1 row	3.57	+5.08	-0.91	+1.67	18.01
	4μ×2 rows	3.64	+4.03	+0.63	+1.27	50.99
LRU	6μ×2 rows	4.82	+8.15	-0.52	+2.74	533.19
=9.50%	2μ×3 rows	2.56	+2.51	-0.11	+0.73	24.13
	4μ×3 rows	4.76	+7.22	-0.56	+2.32	2983.56
	6μ×2 rows†	6.05†	+12.55†	-7.57†	+5.28†	531.66†
DES	4μ×1 row	4.85	+3.53	-0.62	+1.07	15.00
73%	6μ×1 row	6.04	+5.83	-0.87	+1.92	22.25
util.	8μ×1 row	6.48	+7.49	-0.58	+2.52	28.64
	4μ×2 rows	6.28	+6.06	-0.37	+1.94	51.32
LRU	6μ×2 rows	6.76	+8.42	-0.50	+2.82	180.98
=8.30%	2μ×3 rows	5.70	+5.37	-0.54	+1.60	51.71
	4μ×3 rows	6.79	+7.76	-0.62	+2.49	1764.35
	6μ×2 rows†	7.93†	+17.24†	-4.08†	+6.16†	144.15†

lay impact-restricting measures are extremely effective. The leakage reduction is smaller at higher utilization because of lower availability of whitespace (which is favorable to leakage reduction with our experimental setup), and because of fewer opportunities to move cells across rows.

We observe that dynamic power increases with increase in wirelength. In our dynamic power measurement setup, we assume an activity factor of 0.10 on the primary inputs, and activity factors at all other nodes are computed by the power estimation tool [11] using logic propagation. We find that wires contribute $\sim 28\%$ of total dynamic power for our testcase DES, and $\sim 35\%$ for our two AES testcases. Because most of the dynamic power is ascribed to the cells, dynamic power increase is under 2.82% even when wirelength increases by up to 8.42%. We expect the wirelength and dynamic power increases to be lower when our leakage reduction flow is implemented in the placement tool instead of a post-route incremental optimization as we have done.

We consider the leakage reduction results encouraging. Unfortunately, the leakage reduction upper bound is small in our experiments. This is because the placement tool we use inserts fillers next to most cells in the design, and fillers, which have dummy polys that increase pitch, are leakage-favorable in our process. We expect much larger leakage reduction if: (1) fillers do not contain dummy polys as in $90nm$ technologies, (2) the process has an opposite linewidth change as ours, i.e., dense lines print narrower than nominal, or (3) the placer does not create whitespace (with fillers inserted in it) on both sides of most cells in the design.

II.D Aberration-Aware Timing Analysis

Recent studies of lens aberration control have focused on measurement systems [158, 61], pattern sensitivity of aberration [187], and lens mounting systems to compensate for the aberration [123]. However, despite these efforts, the impact of lens aberration on CD will be an ever-present barrier to manufacturing yield as minimum design rules are pushed ever closer to fundamental resolution limits. From the design perspective, variations in CD affect the delays, slews, input

capacitances and leakage of a given logic cell. We also observe that the maximum difference in delays of all timing arcs in a cell (*delay skew*) increases significantly due to lens aberration as different MOS devices in the layout are affected differently by aberration.

Progler et al. [147] studied the impact of lens aberration on statistical timing behavior and observed that certain aberration coefficients are associated with large timing error. Orshansky et al. [136] found that spatial gate CD variation leads to a large variation in the raw speed of CMOS logic. Misleading timing results are obtained, which lead to slower and/or malfunctioning circuits, because the simulation of a circuit's behavior ignored the spatial CD information. The systematic variability of gate CD caused by lens aberration can be modeled in order to achieve better performance by way of accurate timing analysis at all stages of physical implementation [134, 135]. However, more accurate analysis of gate delay impact are required as the scaling of lithographic features makes the lens aberration even more complex. In this section, we describe a novel aberration-aware static timing analysis flow that integrates (i) results of lithography simulation to measure CD across the lens field, (ii) SPICE simulation-based library performance characterization that captures variant CD combinations in library cell instances, and (iii) placement information.

The contributions of our work are as follows.

- Using industry OPC recipes, aberration parameters, and design testcases, we show that the variation in timing due to lens aberration can be significant. Over the cells in a $90nm$ foundry library, we observe cell delay (averaged over all timing arcs) to change by $2\% - 8\%$. The maximum difference in delays of all timing arc of a cell (i.e., delay skew) increases significantly.

- We develop a novel aberration-aware timing analysis flow that affords more accurate timing analysis, taking into account the position of the chip in the lens field. It also considers the increase in delay skew caused by aberration.

Additionally, in [94], we have presented a placement methodology that

utilizes aberration in an analytical placement framework to optimize the circuit delay.

II.D.1 Methodology

Our aberration-aware timing analysis flow involves two main steps: (1) constructing timing libraries of all standard cells for different locations in the lens field, and (2) using placement information of the design to compute the location of all cell instances in the lens field, then using this location information to look up appropriate models in the timing library for use with off-the-shelf static timing analysis (STA) tools.

Before describing our analysis flow, we describe two alternative flows and our reasons for not using them. In the first alternative flow, variants of each cell are created such that the CD of all devices in the cell is different for each variant, but the same for all devices in a given variant. A timing library can be created using SPICE models for all the variants. Since all devices in a cell variant have the same CD, we call this library a *cell-level* granularity library. To perform timing analysis on a placed design, lithography simulation is performed to obtain CDs of all devices in all cells. For each cell, the CDs of its devices are averaged, and the closest-matching available cell variant in the timing library is then fed to off-the-shelf STA. However, as CD skews can be large, averaging of device CDs can introduce inaccuracy in the estimated impact of aberration. In other words, the effect of non-uniform CDs is non-uniformity in timing arc delays, rather than average increase or decrease in the delays of all timing arcs. Our experiments have found that the cell-level library-based approach is very inaccurate compared to the approach that we adopt.

The second alternative flow creates *a priori* variants for each cell master, such that there is one variant for every possible assignment of CDs to devices. This means that given any assignment of CDs to devices, an exactly matching, pre-characterized cell variant can be found. After lithography simulation provides CDs of all devices in all cells, a correctly matching variant can be picked for use in

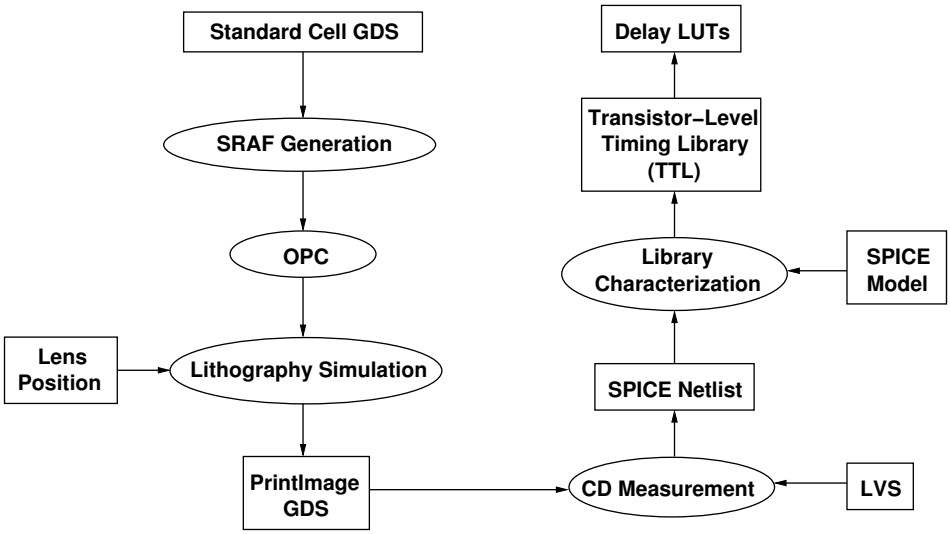


Figure II.19: Aberration-aware timing analysis and its flow.

timing analysis. Though this flow is very accurate, it requires a very large number of cell variants (exponential in the number of devices in the cell); this is infeasible with respect to both characterization time and library size.

In our proposed flow, variants are created for each cell for different lens field locations. Figure II.19 illustrates our timing library construction flow. We begin with standard-cell GDSII files and use *Mentor Graphics Calibre v9.3_5.11* for sub-resolution assist feature (SRAF) generation and model-based OPC. We use Zernike coefficients for eight sampling positions in the lens field from a major chip maker, and compute the other coefficients at 19 different locations with $1.5mm$ stepsize on the field using linear interpolation. Using the post-OPC standard-cell GDS's and Zernike coefficients, we perform lithography simulation at the 19 different field locations with wavelength $\lambda = 193nm$, numerical aperture NA = 0.75, and annular aperture $\sigma = 0.75/0.50$. After lithography simulation, we have 19 PrintImage GDSII results for each standard cell; we then measure the CD of each of the MOS devices in each GDSII result.

Figure II.20(a) shows the PrintImage contour generated by *Mentor Graphics PrintImage* for one device.[7] To measure the CD of the PrintImage contours,

[7]Mentor Graphics PrintImage produces rectilinear contours; our approach, however, is generic

we first take an intersection with the active layer to obtain the contour of the gate. Contours are rectilinearized and split into rectangles in a staircasing fashion. The lengths of all rectangles are then averaged with rectangle widths as weights to compute the CD of the gate (i.e., $CD_{gate} = \sum^n l_i \times w_i / \sum^n w_i$ where n is the number of rectangles into which the contour is split, and l_i and w_i are the lengths and widths of the i^{th} rectangle).

The measured CDs are then used to alter SPICE netlists of standard cells, preparatory to running library characterization. A complication arises because GDSII typically does not have device names, but SPICE netlists only reference devices by device names. We solve this problem by applying LVS (layout vs. schematic) to obtain a mapping between device locations and device names. After modifying the SPICE netlists, we run *Cadence SignalStorm* v4.1 to perform library characterization. Since lens aberration affects different devices in a cell differently, the altered SPICE netlists may no longer have equal CD for all devices. We call our characterized library a *transistor-level timing library* (TTL); it accurately captures the delay skew induced from CD skew while adding manageable complexity to the characterization effort and library size. We also create delay lookup tables (LUTs) to capture the impact of aberration on delay for use in our aberration-aware timing-driven placer as described in [94].

Our test library contains 50 combinational cells. For each we create 19 variants corresponding to the 19 field locations. Library characterization requires approximately 6 hours (wall time) running on 18 CPUs ranging from *Intel Xeon 1.4GHz* to *AMD Opteron 2.2GHz*. We do not create variants for the 13 sequential cells in our library due to large CPU time (estimated at 60 hours on our machines) required by their characterization. We note that the characterization time can be significant but is a one-time task for each process.

enough to be used for arbitrary polygonal contours.

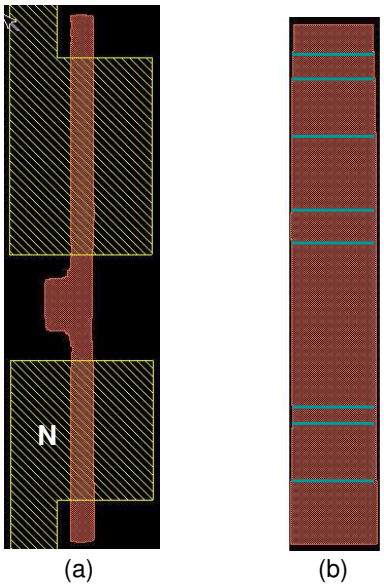


Figure II.20: Polygon generation for CD measurement: (a) result of PrintImage simulation of an inverter, and (b) rectilinearized polygon representation of a gate device in the region N of (a).

Table II.8: Design characteristics of two benchmark circuits.

Design	Utilization (%)	Chip Side (mm)	#Cells	#Nets
AES	60	0.50	17304	17465
JPEG	60	1.41	118321	125036

II.D.2 Experimental Study

In this section, we empirically test our approach on two testcases within a standard industry flow using leading-edge tools. We measure impacts on timing, wirelength, and runtime.

Experimental Setup

We use two designs from *OpenCores* [8] as our testcases. The circuits are synthesized using *Synopsys Design Compiler vW-2004.12-SP3* [11] with tight timing constraints and a set of 63 most commonly used cells in the Artisan TSMC

90*nm* library, then floorplanned in *Cadence SoC Encounter v04.10* [5]. The design characteristics are summarized in Table II.8. We use *Synopsys PrimeTime vZ-2006.12* [14] for static timing analysis. We ignore the delays of wires because they are not differently analyzed or affected by aberration-aware timing analysis.

We compare aberration-aware STA results with traditional STA. For aberration-aware STA, we perform the experiments when there are one, two, and three columns in the lens field. Each *column* has the same width as the die but may contain one or more copies vertically. Die copies in a column suffer identical aberration as explained in Section II.A.2 and will have similar delays. Typically, there are 1-3 columns in a lens field depending on the reticle size, lens reduction factor, and die size [144]. For traditional STA, we assume the worst-case aberration over the 19 locations in the lens field (i.e., the aberration location that yields the worst delay).[8]

Experimental Results

Table II.9 compares the results of traditional STA and aberration-aware STA for our two testcases. From the results we observe that traditional STA is pessimistic with respect to aberration-aware STA. The difference decreases as the number of columns increases. Therefore, it may be worthwhile to add aberration awareness to STA for high-frequency microprocessors and other larger designs which span a large part of the field. Our timing analysis also facilitates calculation of aberration-aware slacks at all nodes in the timing graph. These more accurate slacks can be used in optimizations such as power minimization techniques that exploit timing slack.

[8]Traditional timing analysis used in real-world design flows is likely to be more pessimistic than assumed here. We worst-case by assuming the lens field location that yields the worst-case circuit delay. On the other hand, we expect real-world standard-cell characterization to assume aberration-induced variation that causes the worst cell delay.

Table II.9: Circuit delay reported by traditional STA and aberration-aware STA.

Circuit	Traditional STA Delay (ns)	#Columns	Aberration-Aware STA	
			Delay (ns)	Δ (ps)
AES	2.845	1	2.790	55
		2	2.827	18
		3	2.840	5
JPEG	3.727	1	3.634	93
		2	3.713	14
		3	3.699	28

II.E Conclusions

RETs enable sub-wavelength lithography and are very effective at controlling manufacturing variations. However, variations cannot be completely eliminated, and as feature size continues to shrink, the magnitude of CD variation as percentage increases. Increased use of restricted design rules (RDRs) [118] has been projected for the upcoming technologies to diminish the variations. However, RDRs incur area and consequently functional yield penalty and are not likely to be used extensively for all designs. Even in the designs that use RDRs, the variations are not expected to be completely eliminated. A substantial fraction of CD variation is systematic and can be modeled to improve the accuracy of analysis and efficacy of optimization.

The polysilicon layer is affected severely by CD variations due to the small geometries on it. On the polysilicon layer, CD refers to the linewidth. Linewidth variation determines the variation in the channel length of the NMOS and PMOS devices and has a direct impact on the delay and power of the design. Device saturation current and device capacitance, both of which affect circuit delay, depend nearly linearly on linewidth. Leakage, on the other hand, exhibits a nearly exponential dependence on linewidth.

This chapter has focused on the systematic linewidth variation arising

from pitch, defocus, and lens aberration. Section II.B presents a leakage estimation methodology that models the pitch- and defocus-dependent systematic components of linewidth variation. In this approach, we analyze a layout to calculate device pitches and use them with defocus and a pre-characterized lookup table to predict printed linewidths and to estimate leakage with increased accuracy. Our defocus-aware, topography-oblivious flow does not rely on an STI CMP simulator and assumes defocus variations to be random. It considers device pitches to predict linewidth and consequently leakage with improved accuracy. The defocus-aware, topography-aware flow uses STI CMP simulation to better predict defocus variation and further improve leakage estimation. Our methodology reduces the spread between leakage estimation at worst and best process corners by over half, and can estimate leakages of individual devices with improved accuracy.

Leakage optimization techniques that rely on leakage estimation of individual cells or devices can benefit from the defocus-aware leakage estimation flow. We enhance the previously proposed linewidth biasing methodology that relies on leakage estimation of individual cells to determine the order in which cells are biased. Defocus-aware linewidth biasing has larger leakage reductions than traditional linewidth biasing by $2\% - 7\%$ on our testcases.

In Section II.C, we have proposed a detailed placement approach that arranges cells in standard-cell rows and redistributes whitespace, such that the leakage of the cells is minimized. In doing so, the optimization attempts to minimize the leakage of the cells that offer the most leakage savings, such as lower V_{th} cells and smaller cells since their leakage is most affected by context. We fix the timing critical cells and their interconnects to minimize timing impact and run the optimization over progressively increasing window sizes to minimize wirelength increase. The optimization considers all feasible ways to distribute the cells in the available rows and a TSP-based optimizer places the cells in each row. We assess the TSP-based optimizer to return near-optimal solution quality. And, because all feasible ways to distribute the cells in rows are considered, our final placement in the window is near-optimal. Our results indicate leakage reduction to be in the range of 5%-7% for 7%-8% wirelength increase with negligible delay impact.

While the dynamic power increase is not negligible, for designs and technologies in which leakage is a significant fraction of total power, the leakage savings from our technique will exceed the dynamic power increase. We hypothesize that in technologies in which fillers do not contain dummy polys or in which the process response to pitch variations is opposite to ours, higher leakage reductions would be attained.

In Section II.D, we show that lens aberration affects the linewidths and can impact the delay of many cells by high single-digit percentages. We propose an accurate aberration-aware timing analysis flow to reduce the guardbanding done today in timing analysis. Our approach is layout-aware, and finds the location of each cell in the lens field. Based on the location, timing information that considers aberration is applied to the cell. Our approach is non-obtrusive to existing design tools and utilizes standard STA tools with cell libraries that account for lens aberration. For two benchmark designs in $90nm$ technology, AberrSTA achieves an average guardband reduction of 2.2% in minimum clock cycle time.

II.F Acknowledgments

This chapter is in part a reprint of:

- A. B. Kahng, S. Muddu and P. Sharma, "Defocus-Aware Leakage Estimation and Control," to appear in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

- A. B. Kahng, S. Muddu and P. Sharma, "Defocus-Aware Leakage Estimation and Control," *Proc. International Symposium on Low Power Electronics and Design*, 2005, pp. 263 – 268.

- A. B. Kahng, C.-H. Park, P. Sharma and Q. Wang, "Lens Aberration Aware Timing-Driven Placement," *Proc. Design Automation and Testing in Europe*, 2006, pp. 890 – 895.

- A. B. Kahng, S. Muddu and P. Sharma, "Detailed Placement for Leakage Reduction using Systematic Through-Pitch Variation," *Proc. International Symposium on Low Power Electronics and Design*, 2007, to appear.

I would like to thank my coauthors Swamy Muddu, Chul-Hong Park, Dr. Qinke (Eric) Wang, and Prof. Andrew B. Kahng.

III

STI Stress-Aware Analysis and Optimization

III.A Introduction

At the $65nm$ process node and beyond, it is evident that stress- and strain-based techniques for mobility improvement will be crucial [173]. Enabling progress has been made in the manufacturing process [80, 139, 140, 193, 114, 137, 49, 194, 170] and TCAD (modeling and simulation) [16, 12]. However, stress has not yet been exploited by layout optimizations to improve design performance. In this work, we present a new methodology that combines detailed placement and active-layer fill insertion to exploit STI stress for IC performance improvement. Our methodology begins with process simulation of a production $65nm$ STI technology, from which we generate mobility and delay impact models for STI stress. We develop STI stress-aware SPICE modeling and simulation of critical paths, and finally perform timing-driven optimization of STI stress in standard-cell designs, using detailed placement perturbation to improve PMOS performance and active-layer fill insertion to improve NMOS performance.

Several stress modulation techniques are employed to improve delay and power of CMOS devices. Examples of these techniques are: SiGe stress from underneath the channel, embedded SiGe from the source and drain [80, 139, 140],

single and dual stress liners [193, 114], stress memorization [137, 49], and hybrid orientation [194, 170].

STI stress is the stress that is exerted by STI wells on device active regions, and it affects the performance of CMOS devices. STI is an important and well-studied stress source that has not been fully exploited until now in design quality improvement. Specifically, the dependence of stress on the STI width (STIW) has been neglected until now in circuit-level analyses and optimizations. Several optimizations have been developed to reduce STI stress (e.g., [60, 113, 125]) but they typically fail to completely eliminate layout-dependent STI stress effects. STI usually exerts a compressive stress along the channel (i.e., the current flow direction), which improves PMOS device mobility. NMOS is in general complementary to PMOS in terms of how it is affected by stress, and its mobility degrades because of STI stress. Device mobility increase corresponds to speed increase. Hence, it is possible to utilize STI to improve performance.

Stress induced by STI was analyzed by Gallon et al. [64]. Miyamoto et al. [125] have provided layout-dependent stress analysis of STI. The work of Moroz et al. [126, 127] is significant for indicating possible ways to enhance performance using STI stress; however, no circuit-level optimizations are presented. Recently, Tsuno et al. [177] have shown from $65nm$ silicon data that STI width-dependent stress can impact drive current by up to 10%. With respect to the current body of knowledge: (1) models are still needed to relate stress due to the STI width effect to transistor mobilities, and (2) there is still a lack of available stress optimization methods. A fundamental research goal is to develop novel and efficient simulation, modeling, analysis, and optimization methods to support next-generation stress-aware design automation technology.

Table III.1 shows the impact of STIW on rise and fall delays (averaged over all timing arcs) of several $65nm$ standard cells using the models developed in our work. Impact of placement on STI width and consequently on rise (R-Delay) and fall (F-Delay) delays for several cells are presented. For each cell in the table, three instances of it are placed with different spacings between them, and the delay of the center instance is reported. In Table III.1, *Spacing* is the spacing between

Table III.1: Impact of STI width on performance of several standard cells.

Cell	Space (nm)	PMOS $STIW_L$ (nm)	PMOS $STIW_R$ (nm)	NMOS $STIW_L$ (nm)	NMOS $STIW_R$ (nm)	R-Delay (ps)	F-Delay (ps)
INV0	0um	140	140	110	110	27.27	21.96
	5um	5140	5140	5110	5110	23.65	23.70
BUFD0	0um	140	140	125	125	45.56	46.11
	5um	5140	5140	5125	5125	43.84	43.53
NR2D0	0um	140	140	110	110	51.12	23.06
	5um	5140	5140	5110	5110	42.77	24.69
ND2D0	0um	140	140	110	110	29.63	35.36
	5um	5140	5140	5110	5110	25.77	38.81

cells, and $PMOS\ STIW_L$ ($NMOS\ STIW_L$) and $PMOS\ STIW_R$ ($PMOS\ STIW_R$) are the STI widths next to the left and right sides of positive active (negative active) regions of the center cell. It is possible to both speed up and slow down cells by controlling the STIW and, thereby, the stress that is applied to a cell. In particular, larger STI width will generate more stress in neighboring transistors.

In this chapter, we propose placement perturbation and the insertion of active-layer fills to control the STI width in a performance-driven manner. The proposed active-layer fill insertion and placement perturbation do not require additional process steps or add complications to existing resolution enhancement techniques. Active-layer fill insertion is a standard process step that is performed in all designs to control active-layer density. Placement perturbation yields a new legal placement. We ensure that the design is design rule correct after we perform these two steps.

The remainder of this chapter is organized as follows. In the next section, we briefly describe our STI width-dependent stress modeling and present our models. A detailed description of our stress modeling approach is presented in [100]. In

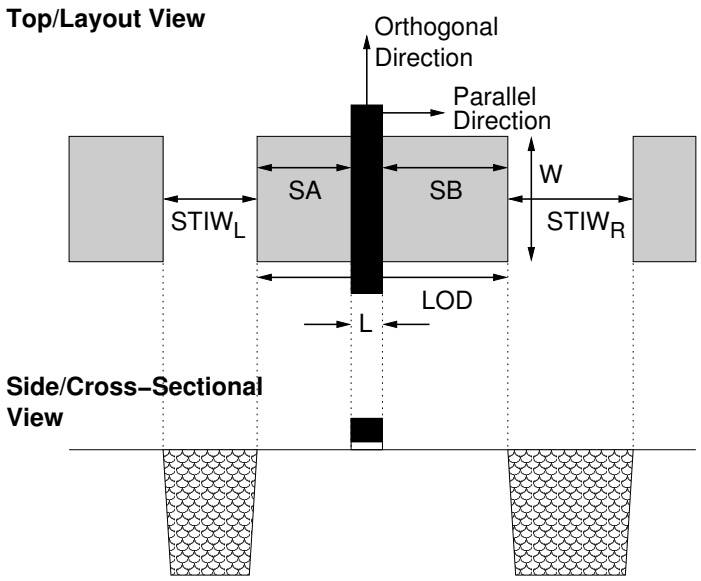


Figure III.1: Various stress-related layout parameters. Parallel and orthogonal distances with respect to a transistor are also indicated.

Section III.C, we describe our STI stress-aware timing analysis flow. Section III.D gives the details of our optimization methodology that utilizes placement perturbation and fill insertion. In Section III.E, we present our experimental results. Finally, we conclude the chapter in Section III.F.

III.B Modeling of STI Width-Dependent Stress

In this section we briefly describe our stress modeling approach; details are presented in [100]. The popularly used BSIM SPICE model (Version 4.3 and higher) contains an explicit STI model. However, only the impact of the distances from transistor channel to the STI boundaries are modeled. In Figure III.1 these distances are labeled as *SA* and *SB*; collectively, these parameters capture the length of diffusion (LOD). The dependency on STIW is not present in the BSIM4 model. Our simulations, as well as simulations and data in the literature [126, 127, 177] show that STIW impact cannot be neglected.

Our objective is to isolate and correct for the impact of STIW, in a

manner that can be applied on top of existing BSIM4 stress modeling. Using 2D TCAD simulations, we have developed the models given in Equations III.2 and III.4 to capture the STIW effect in the parallel direction. The LOD parameter still appears in the equation, as the STIW impact differs according to the value of LOD. Also, for purposes of this work, we do not require or discuss STIW impact modeling in the orthogonal direction, as the STI width effects are blocked in the orthogonal directions by active regions that are typically inserted in standard cells under power and ground lines.

We develop STI width models by curve fitting to data derived from TCAD simulations. Towards this, we simulate CMOS device structures up to the gate deposition step using the *Synopsys Sentaurus 2005.12* [17] process simulator. We perform simulations over combinations of the following four parameters: SA, SB, $STIW_L$ (width of STI on the left side of the active region), and $STIW_R$. At the end of TCAD simulations, we obtain stress values in Pascals and calculate the mobility impact using [164]. The NMOS equation is given as:

$$MOB_{L,R} = \zeta + (1 - (STIW_{L,R}/2)^{\alpha})/S\{A, B\}^{\beta} \tag{III.1}$$

$$MOB = [MOB_L * MOB_R]^{0.26} \tag{III.2}$$

In Equation III.2, MOB is the mobility multiplier and can be implemented as described in [146]. Parameters L and R indicate left and right directions with respect to the channel. The equation states that the final mobility multiplier (i.e., MOB) is the product of the mobility multipliers from the left and right directions (i.e., MOB_L and MOB_R). The PMOS equation is given as:

$$MOB_{L,R} = \zeta + ((STIW_{L,R}/2)^{\alpha})/S\{A, B\}^{\beta} \tag{III.3}$$

$$MOB = [MOB_L * MOB_R]^{0.14} \tag{III.4}$$

The various parameters of the NMOS and PMOS models are given in Table III.2.

Table III.2: Model parameter table

	ζ	α	β
NMOS	1.03	0.076	0.48
PMOS	0.49	0.48	0.57

Analyses and optimizations proposed in the rest of the chapter are not tethered to the models developed in this section, and can be used with other models after appropriate modifications. For example, there are known STI processes that induce tensile instead of compressive stress. This may be due to STI trench height, and material and thermal processing differences, such as HDP (High Density Plasma) CVD (Chemical Vapor Deposition) as used in [55]. Such an STI process as in [55] will have reversed impacts on NMOS and PMOS, and our analysis and optimization methodologies will require appropriate modifications. Furthermore, our models show monotonic response with respect to the STI proximity and widths. In general, the models may be non-monotonic and could require different optimization algorithms.

III.C Stress-Aware Timing Analysis

In this section we describe our STI stress-aware timing analysis methodology. We adapt the traditional SPICE-based timing analysis flow to consider stress induced by STI widths.

III.C.1 Traditional SPICE-Based Timing Analysis

Cell-level static timing analysis (STA) tools such as *Synopsys PrimeTime* [14] offer a good tradeoff between accuracy and analysis speed. Full designs or their blocks are typically analyzed and signed off with circuit-level STA. However, if greater accuracy is desired, SPICE-based analysis, which has better accuracy but substantially slower analysis speed, is employed. Since running full-chip SPICE

analysis is not feasible, critical paths are first identified with static timing analysis and then simulated with SPICE.

A typical netlist input to SPICE is layered into the following three tiers:

- *Device-level models* which contain transistor parameters in the form of coefficients of functions defined in BSIM or equivalent formats. Device-level models allow output waveforms for PMOS and NMOS devices to be simulated.

- *Cell-level netlists* which describe the connectivity of the devices that comprise individual cells. Cell-level netlists instantiate device-level models and allow SPICE to simulate waveforms at the outputs of cells in the library when subjected to a stimulus.

- *Critical path netlists* which describe the connectivity between the cells for each critical path. Critical path netlists instantiate cell-level netlists and can be simulated to calculate the delays of the critical paths.

As noted above, stress-induced device mobility change is determined by (1) the separation between the gate and the active edges, and (2) the size of the STI region that surrounds the active region of the device. Fortunately, the separation between gate and active edges is fixed when the cells are designed, and the contribution of this separation to stress and mobility can be modeled at the cell level. Specifically, in the BSIM 4.3.0 device-level models, stress parameters *SA*, *SB*, and *SD* have been introduced to model the stress effect as a function of gate and active edge separation.[1] In cell-level netlists these parameters are passed with the instantiation of the device-level models. Cell-level netlists are used in library characterization to generate gate-level timing models for use in STA. An example of device-level instantiation with stress parameters is shown in Figure III.2.

The stress effect due to STI width is not modeled primarily for the following two reasons:

[1]Parameters SA and SB are illustrated in Figure III.1. Parameter SD is used in the context of fingered devices to measure the spacing between the fingers; more details are available in the BSIM manual [2].

```
.subckt  INVX1  A Z
 MM1 D G S B  NCH SA=0.2u SB=0.2
 MM2 D G S B  PCH SA=0.19u SB=0.19u
   .
   .
.ends

.model  NCH    NMOS (
 *Other stress parameters defined
   .
   .
)
```

Figure III.2: Instantiation of device-level models in a standard-cell SPICE netlist. The parameters added in BSIM 4.3.0 to partially model stress are shown in **bold**.

- STI width is determined by the placement of the cells, so that stress effect due to STI cannot be captured in library characterization. A new methodology that analyzes a placed design and annotates STI width information for use in timing analysis is required.
- Stress effect due to STI is of smaller magnitude than gate and active edge separation.

III.C.2 STI Stress-Aware Timing Analysis

Our approach analyzes the placement of a design and the standard-cell layouts to calculate the STI widths for all critical cells in the design. The STI widths are then passed as parameters which are used in the models developed in the previous section.

We modify the critical-path netlist to pass the parameters PL, PR, NL, NR to the cell-level netlists at the cell instantiations as shown in Figure III.3. Parameter PL is the spacing between the boundary of a cell and the neighboring active region to the *left* of its positive active region. Similarly, parameter PR is the spacing between the boundary of a cell and the neighboring active region to the *right* of its positive active region (PRX). Parameters NL, and NR are

```
* Critical path 00001
  X01 N1 N2 INVX1  PL=0.08u PR=4.08u NL=0.06u NR=4.06u
  X02 N2 1 N2 NAND2X1  PL=5.0u PR=5.0u NL=5.0u NR=5.0u
  X03 N3 N4 BUFFX1  PL=2.1u PR=5.0u NL=2.04u NR=5.0u
    .
    .
    .

.subckt  INVX1  A Z
  .param PMOB = Our_PMOS_Model (PL, PR, NL, NR)
  .param NMOB = Our_NMOS_Model (PL, PR, NL, NR)
  MM1 D G S B  NCH SA=0.2u SB=0.2  MOB=NMOB
  MM2 D G S B  PCH SA=0.19u SB=0.19u  MOB=PMOB
    .
    .
    .
.ends

.model  NCH  NMOS (
  *Other stress parameters defined
    .
    .
    .
)
```

Figure III.3: Critical paths instantiate cell-level netlists which instantiate device-level models. Our modifications to the traditional flow to model STI width-dependent stress are shown in **bold**.

similarly defined for *negative* active regions (NRX). The cell-level netlists use these parameters to compute the mobility correction factors using our models.

The PL, PR, NL, NR parameters can be calculated from the placement and a given cell's layouts, specifically, the cell boundary to active spacings. Computation of PL for a cell, which is the spacing between the cell's boundary and the positive active region of the cell to its left, is as follows. The spacing between the cell and its left neighboring cell is found from the placement. The spacing between the positive active region of the neighbor and its cell boundary is found from layout analysis of the neighbor. The two spacings are then added, with correct consideration of the orientations of the cell and its neighbor. Other parameters PR, NL, and NR are calculated similarly. Figure III.4 illustrates the calculation.

We note that our flow needs modifications to work for cells with complex

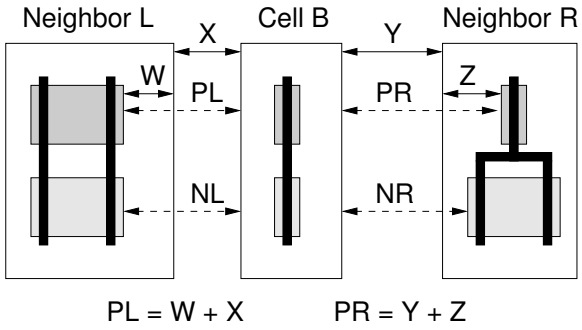


Figure III.4: Calculation of parameters PL, PR, NL, and NR from inter-cell spacings and active to cell boundary spacings.

active shapes such as flip-flops and multiplexers. Active shape complexities include non-rectangular shapes and non-continuous shapes. To model STI stress impact for non-rectangular active shapes, modifications such as those employed by BSIM to handle non-rectangular active may be used. For cells with non-continuous active shapes, devices can be completely shielded from STI width outside the cell and our flow should not alter their mobility. In our analysis and optimization, we focus on the cells with simple active shapes and do not change the mobilities for cells with complex active shapes (i.e., we use traditional analysis and no optimization for them). Fortunately, the most frequently used cells such as inverters, buffers, NAND's, NOR's, AND's, and OR's have simple active shapes, so that we may consider and optimize most cells in our designs.

III.C.3 Alternative Flow

STI stress-aware timing analysis can also be performed by cell-level STA. Toward this end, standard cells in the library can be characterized for different STI width configurations around them. For each standard cell, variants may be created corresponding to each STI width configuration. Given the STI width, models presented in the previous section are used in library characterization. The STI width of a cell in a design can be computed from the placement and the layouts of the standard cells, and can be used to find the variant that has the closest STI

width configuration. The cell can then be bound to the variant in the library, with cell-level STA then run to obtain an STI stress-aware timing analysis.

III.D Timing Optimization

In this section we present our timing optimization methodology. The basic idea exploited in our optimization is that STI widths of devices can be altered to change their mobility and improve performance. Specifically, the alteration involves increasing the STI widths for PMOS devices and decreasing them for NMOS devices. We identify the timing-critical cells and alter their STI widths to improve the circuit performance. In our approach we use the following two knobs to alter the STI widths.

- Placement perturbation. The placement of a layout can be changed to increase or decrease the spacing between neighboring cells, which directly increases or decreases the STI width. Additionally, spacing cells apart can allow fills, for which initially there was insufficient space, to be inserted.

- Active-layer fill (*RX fill*) insertion. Active-layer fills are rectangular dummy geometries inserted on the active (RX) layer primarily to improve planarity after CMP. However, such geometries also reduce the STI width of the devices next to which they are inserted. The STI width after insertion of an RX fill next to a device is the spacing between the active region of the device and the fill.

We now present the details of the above two knobs.

III.D.1 Active Layer (RX) Fill Insertion

Even though RX fills are non-functional geometries, their effect on stress is identical to that of active regions of devices. When inserted next to the active region of an NMOS device, fills substantially reduce the STI width and stress of the device, and consequently improve the performance of the NMOS device. On

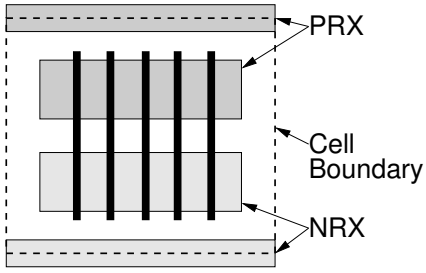


Figure III.5: A generic standard cell with polysilicon, positive active regions, negative active regions, and cell boundary shown.

the other hand, fills inserted next to a PMOS device reduce STI width and stress but consequently degrade the performance. Hence, inserting fill next to the NMOS devices but not next to the PMOS devices of a cell improves performance.

Circuit delay improves when the delay of setup-critical cells is reduced. Thus, we insert rectangular RX fills next to the NMOS devices, to the left and right of the cell. No RX fills are inserted next to the PMOS devices, so that the PMOS remains exposed to a large STI width and stress. The devices closer to the active boundary experience the maximum benefit of this optimization. Since the most frequently used cells in the designs are small, a large fraction of devices in the design can benefit from fill insertion. Our technique can also be employed for hold-time critical cells in the reverse manner, i.e., insert fills next to the PMOS devices but not next to NMOS devices, in order to slow down the cell.

Figure III.5 shows an example standard cell with PRX (active regions for PMOS devices) and NRX (active regions for NMOS devices). As can be seen, active regions exist under the top and bottom cell boundary that completely shield the cell from STI stress effects in the direction orthogonal to the carrier (current) flow direction. Hence, we only apply our optimization in the parallel direction by inserting fill to the right and left of a cell. Figure III.6 illustrates fill insertion for a setup-critical cell; NRX fills are inserted next to the NRX region to reduce stress and *speed up* the NMOS devices. Figure III.7 illustrates the approach for several setup-critical cells in a standard-cell row.

All fills are inserted subject to the design rule constraints (DRCs) and

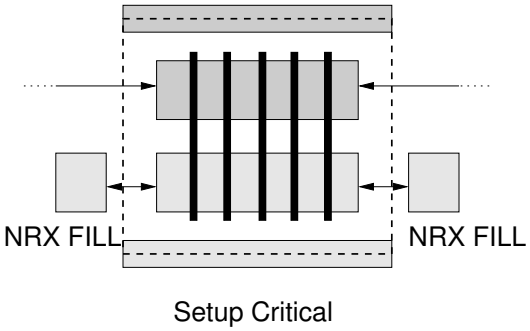


Figure III.6: The generic cell of Figure III.5 optimized with fill insertion for setup criticality.

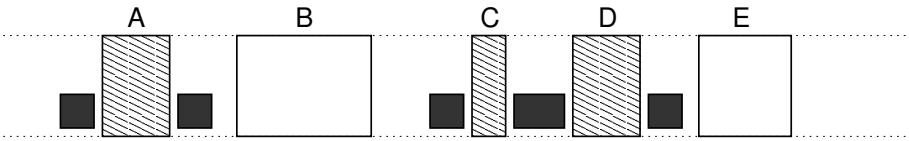


Figure III.7: A row of standard cells after active-layer fill insertion for setup time improvement. Cells patterned with diagonal lines are the setup-critical cells and solidly-filled rectangles are the inserted active-layer fills.

hence do not introduce any DRC violations. We have already noted that no additional mask step is required, and that M1 capacitance impact is likely negligible. Since the fill insertion knob can only decrease STI width, NMOS performance can be improved but PMOS performance can at best be kept constant. However, neighboring cells which have very small spacing and between which fills cannot be inserted can be spaced apart by placement perturbation to allow fills to be inserted.

III.D.2 Intra-Row Placement Optimization

We now present the placement perturbation knob which can increase (decrease) the STI width and improve PMOS (NMOS) performance. Placement of a cell determines its location (and consequently its spacings to neighbor cells) and its orientation. In our optimization, we change the locations of the cells such that spacings are altered but the ordering of cells in a standard-cell row is not affected. Increased spacing next to a cell will increase the STI width and improve the delay of the PMOS devices. However, the delay of the NMOS devices increases with increased spacing. Fortunately, we can utilize our first knob, RX fill insertion, to reduce the NMOS STI width and improve NMOS delay as well. In fact, if the spacing between cells is too small for fill insertion, placement can facilitate fill insertion by creating additional space. The placement perturbation reorganizes the whitespace in a given standard-cell row of the design, without requiring any additional space for the overall layout.

Minimizing delay increase due to wirelength increase. The perturbation of detailed placement from the original placement results in small wirelength change, which can impact wire parasitics and consequently timing. Even though our localized placement perturbations do not significantly affect timing, small changes in the timing of critical paths can affect the minimum clock cycle time. To minimize the timing change of critical paths, we *fix* the cells and nets in the critical paths. Fixed cells are not moved during optimization, and fixed nets are not changed during the ECO routing that is performed after placement opti-

mization. Since the nets in the critical path are fixed, all cells connected to these nets should also be marked as fixed and not moved during optimization. We note that the delay of such nets can marginally change due to the coupling capacitance with neighboring nets, the routing for which may change. We also fix all flip-flops, clock buffers, and clock nets to avoid any impact on the clock tree. Thus, our list of *fixed cells* comprises timing-critical cells, their fanout cells, flip-flops, and clock buffers.

Our intra-row placement optimization attempts to create space on the right and left sides of each timing-critical cell. In the process, a minimum number of cells is displaced to minimize the wirelength impact. Figure III.8 presents the pseudo-code for our intra-row placement optimization. For each timing-critical cell, right and left spacings are increased by functions *createRightSpace* and *createLeftSpace* respectively to attain a spacing of up to S. The spacing, S, may not always be attainable because of the presence of fixed cells and availability of limited space in the row. For the right side, the function *cellsToMoveRight* finds the minimum number of cells to move. Then the function *moveCellsRight* flushes the computed number of cells to the right as much as possible.

Our algorithm sequentially processes critical cells in decreasing order of their criticality. Cells displaced in an iteration to create space are added to the list of fixed cells to lock their placements during succeeding iterations. This can limit the optimization of critical cells processed later in the algorithm. Therefore, we run the algorithm multiple times with increasing value of S. This enhancement allows a fair distribution of whitespace among all critical cells. The experiments reported below increase the value of S from $0.6\mu m$ to $1.8\mu m$ in steps of $0.2\mu m$. We have found that the STI width effect saturates at $1.8\mu m$ and that there is negligible change in stress beyond this value.

Our second enhancement is to perturb the placement of critical cells to balance the space on the right and left sides of them. Since the stress effect decays rapidly with space, it is desirable to have nearly-equal spacings on both sides. We limit the perturbation to $0.6\mu m$ to minimize wirelength and any associated delay increase. The space required to insert RX fill is typically very small and

Input: Placed design; set of timing-critical cells, T; set of fixed cells, F; maximum spacing to create, S

Output: New placement with altered inter-cell spacings

[1] **forall** cells $t \in T$

[1.1] createRightSpace(t)

[1.2] createLeftSpace(t)

createRightSpace(t)

[1] n = cellsToMoveRight(t);

[2] moveCellsRight(t, n);

cellsToMoveRight(t)

[1] i $\leftarrow$ t;

[2] j $\leftarrow$ cellToRightOf(t);

[3] accumulatedSpacing = 0;

[4] cellsToMove = 0;

[5] **while** accumulatedSpacing $\leq S$ **and** $j \notin F$ **and** cellToRightOf(j) $\notin T$

[5.1] accumulatedSpacing += interCellSpacing(i, j);

[5.2] cellsToMove++;

[5.3] i $\leftarrow$ j;

[5.4] j $\leftarrow$ cellToRightOf(i);

[6] return cellsToMove;

moveCellsRight(t, n)

[1] // flush n cells to the right of Cell t towards the right to create space S

[2] F $\leftarrow$ F $\cup$ {n cells to the right of Cell t}

createLeftSpace(t)

[1] // similar to createRightSpace(t)

Figure III.8: Pseudo-code for intra-row placement optimization.

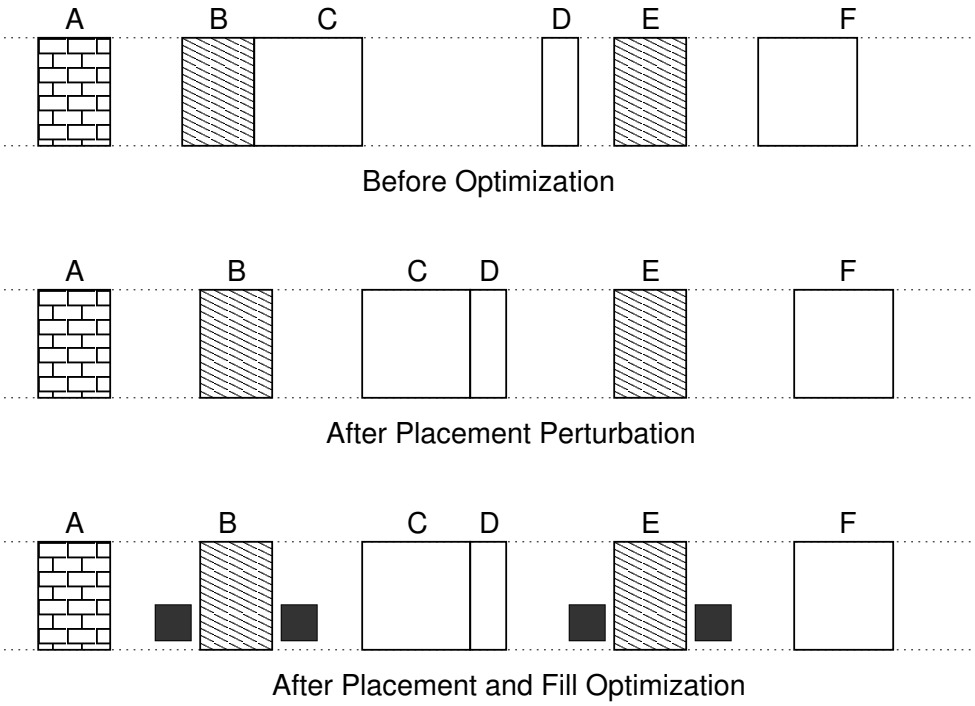


Figure III.9: Placement change and fill insertion for setup-time optimization. A standard-cell row is shown before optimization, after placement perturbation, and after fill insertion. Cells patterned by diagonal lines are the setup-critical cells for which timing is optimized. Fixed cells are patterned with the brick pattern and their placement cannot be changed.

in the $0.2\mu m$ range. Therefore, if the optimization creates any space for PMOS optimization, fill can always be inserted to mitigate the deterioration of NMOS performance. Figure III.9 illustrates placement perturbation and fill insertion for setup-time optimization of a standard-cell row.

While it is possible to perform fill insertion without placement perturbation, we have found the associated performance benefits to be very small. The two knobs complement each other: placement creates space for fill insertion and fill insertion improves the performance of the NMOS devices that are slowed down by placement perturbation.

Our overall STI stress-aware placement and fill optimization flow is as follows.

1. Identify critical paths and critical cells.
2. Perform intra-row placement optimization.
3. Perform ECO routing followed by parasitic extraction.
4. Perform RX fill insertion.
5. Evaluate the optimized layout with STI stress-aware timing analysis.

III.E Experimental Study

We now present our experiments to evaluate the proposed optimization methodology. Our experiments assess the impact of our optimization on the minimum clock cycle time, delays of top critical paths, and final routed wirelength.

III.E.1 Experimental Setup

The details of the testcases used in our experiments are presented in Table III.3. We use *Synopsys Design Compiler vW-2004.12.SP3* [11] for synthesis, *Cadence SOC Encounter (v5.2)* [5] for placement, clock tree synthesis, routing, and parasitic extraction, *Synopsys PrimeTime vW-2004.12.SP2* [14] for cell-level timing analysis, and *Synopsys HSPICE vY-2006.03* [13] for SPICE simulations. For our experiments, we use the 50 most frequently used cells from high-V_{th} and nominal-V_{th} $65nm$ general purpose libraries. SPICE device models and cell netlists were supplied by a foundry. We built our optimizer on top of *OpenAccess API v2.2.4* [7].

III.E.2 Experimental Results

We first compare the proposed stress-aware timing analysis with traditional analysis. Since traditional analysis does not account for STI stress and must correctly analyze for all STI configurations, it is conservative. Traditional analysis is corner-based and uses the worst-case cell delays, which reflect worst-case

Table III.3: Testcases used in experimental validation. *MCT* is the minimum cycle time.

Circuit	Source	#cells	Utilization	MCT (ns)
C5315	ISCAS'85	1,408	82%	0.912
ALU	opencores.org	11,106	78%	4.333
S38417	ISCAS'85	8,514	79%	3.086
AES	opencores.org	21,000	78%	4.738

STI stress effects in addition to worst-case process variations. Worst-case analysis, while correct, leaves valuable performance on the table. Stress-aware timing analysis reduces pessimism in analysis by explicitly accounting for STI stress. We therefore expect stress-aware timing analysis to report circuit delays that are smaller than from traditional analysis.

Table III.4 presents the comparison between traditional timing analysis and stress-aware timing analysis on our four testcases. We study two delay metrics: (1) minimum cycle time (MCT), (2) and *top paths delay* (TPD), which is the sum of the delays of top 100 critical paths. While MCT determines the maximum speed at which the circuits can be run, TPD determines the robustness to variations. We observe that stress-aware analysis reduces MCT by 5.75%, and TPD by 5.28%, on average. We use stress-aware analysis to evaluate our optimization in the remainder of this section.

In Section III.D we presented two optimization knobs: fill insertion and placement perturbation. Although the two techniques complement each other, we evaluate the fill insertion knob separately. As discussed above, placement perturbation without fill insertion is not interesting because it slows down NMOS devices while speeding up PMOS. Table III.5 presents the improvements in MCT and TPD due to fill insertion. Since we optimize several critical paths, TPD is reduced. However, we observe that reductions in MCT and TPD are typically under 1%.

Table III.4: Traditional vs. stress-aware timing analysis.

Circuit	Traditional		Stress-Aware			
	MCT (ns)	TPD (ns)	MCT (ns)	MCT Improv. (%)	TPD (ns)	TPD Improv. (%)
C5315	0.977	87.43	0.915	6.31	81.93	6.29
ALU	1.885	185.50	1.778	5.68	175.24	5.53
S38417	1.068	104.95	1.018	4.68	99.58	5.11
AES	1.739	165.82	1.655	4.83	158.88	4.19

Table III.5: Timing optimization results with fill insertion. *MCT* is the minimum cycle time. *WL* is the wirelength. *TPD* stands for top paths delay and is the sum of the delays of the top 100 critical paths.

Circuit	Original		Fill Opt			
	MCT (ns)	TPD (ns)	MCT (ns)	MCT Improv. (%)	TPD (ns)	TPD Improv. (%)
C5315	0.915	81.83	0.903	1.32	81.35	0.71
ALU	1.778	175.24	1.771	0.39	174.53	0.40
S38417	1.018	99.58	1.010	0.79	99.92	0.39
AES	1.655	158.88	1.651	0.24	158.55	0.21

Table III.6: Timing optimization results with placement and fill insertion. *MCT* is the minimum cycle time. *WL* is the wirelength. *TPD* stands for top paths delay and is the sum of the delays of the top 100 critical paths.

Circuit	Original			Placement & Fill Opt			Reduction		
	MCT (ns)	TPD (ns)	WL (mm)	MCT (ns)	TPD (ns)	WL (mm)	MCT (%)	TPD (%)	WL (%)
C5315	0.915	81.93	17.8	0.879	75.50	17.9	3.97	7.85	+0.67
ALU	1.778	175.24	196.1	1.709	168.14	196.8	3.88	4.05	+0.36
S38417	1.018	99.58	96.4	0.993	97.94	96.64	2.44	1.65	+0.23
AES	1.655	158.88	374.7	1.568	153.21	3.75	5.26	3.56	+0.08

We next evaluate the simultaneous use of the proposed placement perturbation and fill insertion knobs. In addition to comparing MCT and TPD results, we also compare wirelength, which changes because of placement perturbation. After placement perturbation, several nets are left dangling; we perform ECO routing to route them, followed by RC extraction and stress-aware timing analysis to obtain accurate post-optimization MCT and TPD values. Table III.6 presents our results for our four testcases. With negligible increase in wirelength, we observe 4.37% and 5.15% reductions in (stress-aware) MCT and TPD averaged over the testcases *C5315*, *ALU*, and *AES*. The testcase *S38417* demonstrates smaller improvements; we attribute this to the fact that *S38417* is an artificial testcase with over 50% of its cells being flip-flops. To avoid changes to the clock tree, we do not allow our optimization to change the locations of flip-flops, and hence in the *S38417* testcase, we can perturb the placement of fewer cells. Figure III.10 shows the histograms for the delays of top critical paths of the testcase *AES* before and after optimization. Our optimization shifts the delay distribution to the left (lower delay) substantially.

We also attempted optimization of hold-critical paths but found negligible improvement in hold slack for our testcases. This is because stress optimization can

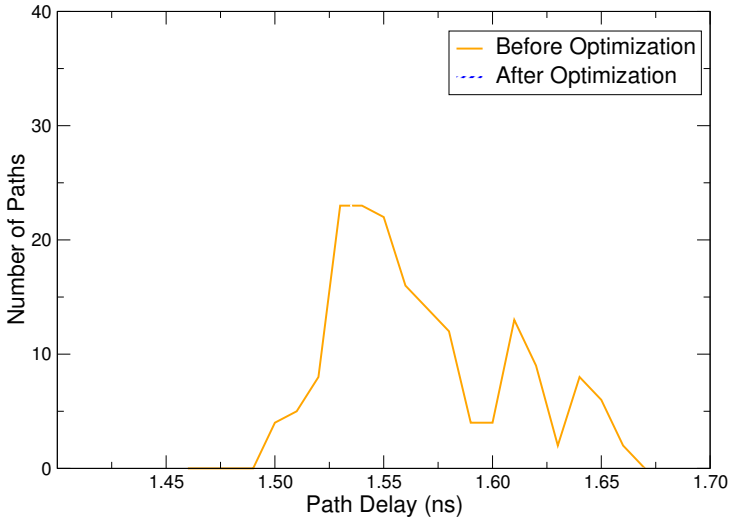


Figure III.10: Path delay histograms for the top 200 critical paths of testcase AES before and after optimization.

only change cell delays by 10%-20% and for hold-critical paths the cell delays are very small. As a result, the change in the delay of hold-critical paths is insignificant with our approach, and traditional delay introduction methods such as insertion of delay elements or wire snaking must be used.

III.F Conclusions

We have conducted TCAD process simulations to generate models that relate the dependence of transistor mobilities to stress induced by STI width. Using our models, we find that the delay of standard cells varies significantly depending on their placement, which affects the widths of the neighboring STI regions. We have proposed an STI width-aware circuit delay analysis flow that uses our models along with placement information. The proposed stress-aware timing analysis technique reduces pessimism in delay analysis. Compared with traditional corner-based analysis, critical-path delays reported by stress-aware analysis are on average 5.75% lower. We have also devised an optimization methodology, based on cell placement perturbation, to create extra space around critical cells; this

is followed by active-layer fill insertion. The proposed optimization flow, while demonstrated with our models, can be adapted to other STI stress models. We have applied the proposed optimization flow on a number of testcases implemented with industry $65nm$ libraries. Our data shows that STI width optimization can improve performance by 2.44% to 5.26% with no area penalty and with negligible increase in wirelength. The proposed optimization can form the basis of circuit optimizations that exploit upcoming stress-engineered transistor technologies in $65nm$ and below processes.

III.G Acknowledgments

This chapter is in part a reprint of: A. B. Kahng, P. Sharma and R. O. Topaloglu, "Exploiting STI Stress for Performance," *Proc. International Conference on Computer-Aided Design*, 2007, to appear.

I would like to thank my coauthors Rasit O. Topaloglu, and Prof. Andrew B. Kahng. Additionally, I would like to thank Prof. Hoong-Joo Lee of Sangmyung University, Korea, and Frank Geelhaar of AMD for useful discussions, and Swamy Muddu for help with the implementation and the testcases.

IV

Enhancing Design Robustness to Gate Length Variations

IV.A Introduction

The focus of this chapter is on leakage power and its variability. As discussed in Section II.B, leakage power has become one of the most critical design concerns for chip designers. Manufacturers face the additional challenge of *leakage variability*: recent data indicates that leakage of microprocessor chips from a single $180nm$ wafer can vary by as much as $20\times$ [29].

Leakage reduction techniques can be divided into two classes depending on whether they reduce *standby* leakage or *runtime* leakage. Standby techniques reduce leakage of inactive devices (i.e., devices that do not switch), while runtime techniques reduce leakage of active devices. Several techniques have been proposed for standby leakage reduction. *Body biasing* or *VTMOS* based approaches [86] dynamically adjust the device V_{th} by biasing the body terminal.[1] *Multi-threshold CMOS (MTCMOS)* techniques [128, 103, 129, 157] use high-V_{th} PMOS and/or NMOS devices to disconnect V_{DD} and/or V_{SS} to the devices in standby mode. *Source biasing*, where a positive (negative) bias is applied in standby state to source terminals of inactive NMOS (PMOS) devices, was proposed in [84]. Other

[1]Body biasing has also been proposed to reduce leakage of active devices [133].

techniques such as use of transistor stacks [195] and input-vector control [79] have also been proposed.

The only mainstream approach to runtime leakage reduction is the multi-V_{th} manufacturing process. In this approach, cells on non-critical paths are assigned a high V_{th} while cells on critical paths are assigned a low V_{th}. [184] presents a heuristic algorithm for selection and assignment of an optimal high V_{th} to cells on non-critical paths. The multi-V_{th} approach has also been combined with several other power reduction techniques [110, 186, 161]. The primary drawback to this technique has traditionally been the rise in process costs due to additional steps and masks. However, the increased costs have been outweighed by the resulting substantial leakage reductions, and multi-V_{th} processes are now standard. A new complication facing multi-V_{th} is the increased variability of V_{th} for low-V_{th} devices. This occurs in part due to random doping fluctuations, as well as worsened drain induced barrier lowering (DIBL) and short-channel effects (SCE) in devices with lower channel doping. The larger variability in V_{th} degrades the achievable leakage reductions of multi-V_{th} and worsens with continued MOS scaling. Moreover, multi-V_{th} methodologies do not offer a smooth tradeoff between performance and leakage power. Devices with different V_{th} typically have a large separation in terms of performance and leakage, for instance a 15% speed penalty with a 10× reduction in leakage for high-V_{th} devices.

The use of longer gate lengths (L_{Gate}) in devices within non-critical gates to reduce runtime leakage was first described in [163]. In that work, *large* changes to gate lengths were considered, resulting in heavy delay and dynamic power penalties. Moreover, cell layouts with significantly larger gate lengths are not layout-swappable with their nominal versions, resulting in substantial ECO overheads during layout. In this chapter, we propose very *small* increases in gate length for non-critical devices. These small increases maximize the leakage reduction since they take full advantage of the SCE and incur only very small penalties in drive current and input capacitance. Technologies at the $90nm$ node and below employ super-halo doping, giving rise to reverse short channel effects (RSCE) that mitigate traditional SCE to some extent. However, we have found the proposed technique to

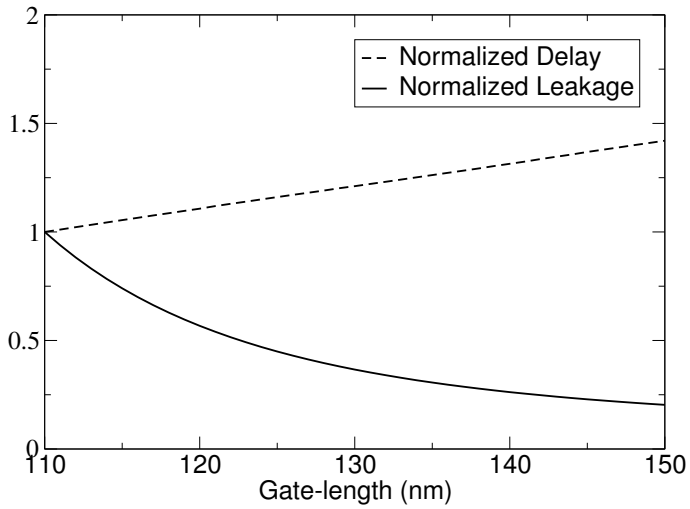


Figure IV.1: Variation of leakage and delay (each normalized to 1.00) for an NMOS device in an industrial 130nm technology.

substantially reduce leakage for the two 130nm and two 90nm industrial processes that we investigated. Recent reports from leading integrated device manufacturers (IDMs) indicate SCE continues to dominate V_{th} roll-off [171] characteristics at the 65nm and 45nm technology nodes [130, 131, 27, 120]. However, we note that the V_{th} roll-off curve must be understood to assess the feasibility of this approach and to determine reasonable increases for gate length.

The variation of delay and leakage with gate length is shown in Figure IV.1 for an industrial 130nm process. Leakage current flattens out with gate length beyond 140nm, making L_{Gate} biasing less desirable in that range. Another major advantage of L_{Gate} biasing is leakage variability reduction. Since the sensitivity of leakage to gate length reduces with increased gate length, a fixed absolute level of variability in gate length (e.g., $\pm 3nm$) translates to reduced relative variability in leakage. We use the terms *gate length biasing* and L_{Gate} *biasing* interchangeably to refer to the proposed technique. We use the phrase *biasing a device* to refer to increasing the gate length of the device slightly.

We also assess the costs and benefits of transistor-level L_{Gate} biasing (TLLB). Since different transistors control different timing arcs of a cell, TLLB can individually modify delays of different timing arcs. Our hypothesis is that asym-

metry in timing criticality of different timing arcs of a cell instance in a circuit, and that of rise and fall transitions, can be exploited by TLLB to yield significant leakage savings. [185, 162, 104] proposed transistor-level V_{th} assignment for leakage power reduction. Our approach uses L_{Gate} biasing instead of V_{th} assignment and is otherwise similar to [185]. The major disadvantage of TLLB (and V_{th} assignment) is the increase in library size and its characterization time.

Contributions of our work include the following.

- A leakage reduction methodology based on less than 10% increase in drawn L_{Gate} of devices.
- A thorough analysis of potential benefits and caveats of such a biasing methodology, including implications for lithography and process variability.
- Experiments and results showing potential benefits of our gate length biasing methodology in different design scenarios such as dual-V_{th}.
- Assessment of the impact of L_{Gate} biasing on the choice of threshold voltages.
- Adaptation of Lagrangian relaxation-based sizing proposed in [44] to L_{Gate} biasing, and quantification of suboptimality using synthetic testcases for which upper bounds on minimization objectives can be easily derived.

The organization of this chapter is as follows. In the next section, we describe the proposed L_{Gate} biasing methodology for leakage reduction. Section IV.C applies L_{Gate} biasing at the transistor level by allowing transistors within a cell to be selectively biased. Section IV.D gives experiments and results for validation of the proposed ideas. It also analyzes the potential manufacturing and process variation implications of biasing gate lengths. In Section IV.E, we empirically study the impact of the availability of biasing on threshold voltage selection. Section IV.F presents a potentially better optimization based on Lagrangian relaxation. Section IV.G sketches an approach to evaluate the optimization quality by quantifying the suboptimality on synthetic testcases. Finally, Section IV.H concludes the chapter.

IV.B Cell-Level Gate Length Biasing

In this section we describe the proposed cell-level L_{Gate} biasing (CLLB) methodology. Our approach extends a standard-cell library by adding biased variants to it. We then use a leakage optimization approach to incorporate slower, low-leakage cells into non-critical paths, while retaining faster, high-leakage cells in critical paths.

IV.B.1 Library Generation

We generate a restricted library composed of variants of the 25 most commonly used cells in our testcases.[2] For each of these identified cells, we add a *biased variant* in which all devices within the cell have the biased gate length. We consider less than 10% biasing because of the following reasons:

- The nominal gate length of the technology is usually very close to or beyond the "knee" of the leakage vs. L_{Gate} curve (shown in Figure IV.1) which arises due to SCE. For large bias, the advantage of super-linear dependence of leakage on gate length is lost. Moreover, dynamic power and delay both increase almost linearly with gate length. Therefore, small biases give more "bang for the buck".

- From a manufacturability point of view (discussed later in Section IV.D.2), having two prevalent pitches (which are relatively distinct) in the design can harm printability properties (i.e., size of process window). Therefore, we retain the same poly pitch as the unbiased version of the cell. On biasing, there is a small decrease in spacing between gate-poly geometries. However, minimum poly spacing is larger than gate length, so the process window, which is constrained by the minimum resolvable dimension, tends to improve (i.e., enlarge).

[2] We first synthesize our testcases with the complete Artisan TSMC 130nm library to identify the most frequently used cells.

- Small biases allow pin-compatibility with the unbiased version of the cell. This is very important to ensure that multi-L_{Gate} optimizations can be done post-placement or even after detailed-routing without ECOs. In this way, we retain the layout transparency that has made multi-V_{th} optimization so adoptable within chip implementation flows. Small biases can be realized during RET by using biasing information in RET recipes, and biased cells have identical physical attributes for use in design as their unbiased counterparts.

For the SPICE models we use, the nominal gate length of all transistors is $130nm$. In our approach, all transistors in a biased variant of a cell have a gate length of $138nm$. We choose $138nm$ as the biased gate length because it places the delay of the low-V_{th}-biased variant between those of the low-V_{th}-nominal gate length variant and the nominal-V_{th}-nominal gate length variant. Larger bias can lead to larger per-cell leakage saving at a higher performance cost. However, in a resizing setup (described below) with a delay constraint, the leakage benefit over the whole design can decrease as the number of instances that can be replaced by their biased version is reduced. Larger or smaller biases may produce larger leakage reductions for some designs. Libraries, however, are not design-specific, and a biased gate length that produces good leakage reductions for all designs must be chosen. We have found the above-mentioned approach for choosing the biased gate length to work well for all designs. We note that the value of $138nm$ is highly process-specific and is not intended to reflect the best biased gate length for all $130nm$ processes. We have previously discussed biasing at finer levels of granularity (i.e., having multiple biased gate lengths and independently biasing devices within a cell) in [77]. However, we did not find significant leakage savings above cell-level biasing.[3]

[3]We have been informed that a major U.S. semiconductor manufacturer has started to offer its customers a cell-wise L_{Gate} biased variant of its 90nm cell library with a 6nm bias. Also, a recent paper by Texas Instruments describes a very similar approach used by them [152]. This not only reinforces the viability of the methodology we describe, but also suggests that our use of an 8nm bias for a 130nm cell library provides a practically relevant testbed.

An important component of our leakage reduction methodology is layout and characterization of the dual-L_{Gate} library. Since we investigate very small biases to the gate length, the layout of the biased library cell does not need to change except for a simple automatic scaling of dimensions. Moreover, since the bias is smaller than the minimum layout grid pitch, design rule violations do not occur. Of course, after the slight modifications to the layout, the biased versions of the cell are put through the standard extraction and power/timing characterization processes.

IV.B.2 Optimization for Leakage

We perform standard gate sizing (gate width sizing) prior to L_{Gate} biasing using *Synopsys Design Compiler v2003.06-SP1* [11]. Since delay is usually the primary design goal, we perform sizing to achieve the minimum possible delay. We use a sensitivity-based *downsizing* (i.e., begin with all nominal cell variants and replace cells on non-critical paths with biased variants) algorithm for leakage optimization. In our studies, we have found downsizing to be significantly more effective at leakage reduction than *upsizing* (i.e., begin with all biased variants in the circuit and replace critical cells with their nominal-L_{Gate} variants) irrespective of the delay constraints. An intuitive rationale is that upsizing approaches have dual objectives of delay and leakage during cell selection for upsizing. Downsizing approaches, on the other hand, only downsize cells that do not cause timing violations and have the sole objective of leakage minimization. We note that an upsizing approach, however, may be faster when loose delay constraints are to be met since very few transistors have to be upsized. However, delay is almost always the primary design goal and loose delay constraints are rare.

A timing analyzer is an essential component of any delay-aware power optimization approach; it is used to compute delay sensitivity to biasing of cell instances in the design. For an accurate yet scalable implementation, we use three types of timers that vary in speed and accuracy.

- *Standard static timing analysis (SSTA).* Slews and actual arrival times (AATs)

are propagated forward after a topological ordering of the circuit. Required arrival times (RATs) are back-propagated and slacks are then computed. Slew, delay and slack values of our timer match exactly with *Synopsys PrimeTime vU-2003.03-SP2* [14] and our timer can handle unate and non-unate cells.[4]

- *Exact incremental STA (EISTA).* We begin with the fan-in nodes of the node that has been modified. From all these nodes, slews and AATs are propagated in the forward direction until the values stop changing. RATs are back-propagated from only those nodes for which the slew, AAT or RAT has changed. Slews, delays and slacks match exactly with SSTA.

- *Constrained incremental STA (CISTA).* Sensitivity computation involves temporary modifications to a cell to find change in its slack and leakage. To make this step faster, we restrict the incremental timing calculation to only one stage before and after the gate being modified. The next stage is affected by slew changes and the previous stage is affected by the pin capacitance change of the modified gate. The ripple effect on other stages farther away from the gate (primarily due to slew changes[5]) is neglected since high accuracy is not critical for sensitivity computation.

We use the phrase "downsizing a cell instance" (or node) to mean replacing it by its biased variant in the circuit. In our terminology, s_p represents the slack on a given cell instance p, and s'_p represents the slack on p after it has been downsized. ℓ_p and ℓ'_p denote leakages of cell instance p before and after downsizing, respectively. P_p represents the sensitivity associated with cell instance p and captures the leakage reduction achieved on biasing it per unit slack decrease at its output. Several previous works have used sensitivity-based greedy approaches for circuit optimization. Fishburn et al. [62] use a sensitivity function that captures

[4]Delay values from our timer match with PrimeTime only under our restricted use model. Our timer does not support several important features such as interconnect delay, hold time checks, false paths, multiple clocks, etc.

[5]There may be some impact from coupling-induced delay also, as the arrival time windows can change; we ignore this effect.

the change in gate delay per unit change in gate width to perform gate width sizing. Sirichotiyakul et al. [161] use a sensitivity function to perform power optimization by V_{th} assignment. Their sensitivity function is the ratio of leakage change and delay change, weighted to capture timing criticality of paths. Our sensitivity function is defined as:

$$P_p = \frac{\ell_p - \ell'_p}{s_p - s'_p} \tag{IV.1}$$

The pseudo-code for our leakage optimization implementation is given in Figure IV.2. The algorithm begins with SSTA and initializes slack values s_p in Line 1. Sensitivities P_p are computed for all cell instances p and put into a set S in Lines 2-5. We select and remove the largest sensitivity P_{p^*} from the set S and continue with the algorithm if $P_{p^*} \geq 0$. In Line 11, the function $SaveState$ saves the gate lengths of all transistors in the circuit as well as the delay, slew and slack values. The cell instance p^* is downsized and EISTA is run from it to update the delay, slew and slack values in Lines 12-13. Our timing libraries capture the effect of biasing on slew as well as input capacitance, and our static timing analyzer efficiently and accurately updates the design to reflect the changes in delay, capacitance and slew due to the downsizing move. If there is no timing violation (negative slack on any timing arc) then this move is accepted, otherwise the saved state is restored. If the move is accepted, we also update sensitivities of node p^*, its fan-in nodes and its fan-out nodes in Lines 17-21. The algorithm continues until the largest sensitivity becomes negative or the size of S becomes zero. Function $ComputeSensitivity(q)$ temporarily downsizes cell instance q and finds its slack using CISTA. Since high accuracy is not critical for sensitivity computation we choose to use CISTA which is faster but less accurate than EISTA. Table IV.1 shows a comparison of leakage and runtime when EISTA and CISTA are used for sensitivity computation.[6]

[6]The results correspond to transistor-level gate length biasing which uses the same optimization algorithm as cell-level biasing.

procedure $L_{Gate}Biasing$

[1] Run SSTA to initialize s_p $\forall$ cell instances, p
[2] $S \leftarrow \{\}$
[3] **forall** cell instances, p
[4] $P_p \leftarrow ComputeSensitivity(p)$
[5] $S \leftarrow S \cup P_p$
[6] **do**
[7] $P_{p^*} \leftarrow \mathbf{max}(S)$
[8] **if**$(P_{p^*} \leq 0)$
[9] **exit**
[10] $S \leftarrow S - \{P_{p^*}\}$
[11] $SaveState()$
[12] Downsize cell instance p^*
[13] $EISTA(p^*)$
[14] **if**$(TimingViolated())$
[15] $RestoreState()$
[16] **else**
[17] $N \leftarrow p^* \cup$ fan-in and fan-out nodes of p^*
[18] **forall** $q \in N$
[19] **if**$(P_q \in S)$
[20] $P_q \leftarrow ComputeSensitivity(q)$
[21] Update P_q in S
[22] **while**$(|S| > 0)$

Figure IV.2: Pseudocode for cell-level gate length biasing for leakage optimization. Procedure *ComputeSensitivity* is defined in Figure IV.3.

procedure *ComputeSensitivity*(q)

[1] $old\_slack \leftarrow$ Slack on cell instance q

[2] $old\_leakage \leftarrow$ Leakage of cell instance q

[3] *SaveState*()

[4] Downsize cell instance q

[5] $CISTA(q)$

[6] $new\_slack \leftarrow$ Slack on cell instance q

[7] $new\_leakage \leftarrow$ Leakage of cell instance q

[8] *RestoreState*()

[9] **return** $(old\_leakage - new\_leakage)/(old\_slack - new\_slack)$

Figure IV.3: Pseudocode for the *ComputeSensitivity* procedure.

Table IV.1: Comparison of leakage and runtime when EISTA and CISTA are used for sensitivity computation.

Circuit	Leakage (mW)		CPU (s)	
	EISTA	CISTA	EISTA	CISTA
s9234	0.0712	0.0712	4.86	2.75
c5315	0.3317	0.3359	24.18	14.99
c7552	0.6284	0.6356	55.56	43.79
s13207	0.1230	0.1228	33.43	17.15
c6288	1.8730	1.9157	508.86	305.09
alu128	0.4687	0.4857	1122.89	544.75
s38417	0.4584	0.4467	1331.49	746.79

Table IV.2: Asymmetry in delays and slews (transition delays) of various timing arcs within a NAND2X2 standard cell.

Timing Arc	Propagation Delay (ps)	Transition Delay (ps)
A $\rightarrow$ Y $\uparrow$	99.05	104.31
A $\rightarrow$ Y $\downarrow$	73.07	79.12
B $\rightarrow$ Y $\uparrow$	107.20	112.98
B $\rightarrow$ Y $\downarrow$	70.65	76.37

IV.C Transistor-Level Gate Length Biasing

We use the term *timing arc* to denote a pair of input transition (rise or fall) and the resulting output transition (rise or fall). For an n-input gate, there are $2n$ timing arcs.[7] Due to different parasitics as well as PMOS/NMOS asymmetries, these timing arcs can have different delay values associated with them. For instance, Table IV.2 shows the delay values for the same input slew and load capacitance pair for different timing arcs of a NAND2X2 cell from the Artisan *TSMC* 130nm library. Pin swapping is a common post-synthesis timing optimization step to make use of the asymmetry in delays of different input pins. To make use of asymmetry in rise-fall delays, techniques such as P/N ratio perturbations have been previously proposed to decrease circuit delay [24]. We propose to exploit these asymmetries using TLLB to "recover" leakage from non-critical timing arcs within a cell.

IV.C.1 Library Generation

For each cell, our library contains variants corresponding to all subsets of the set of timing arcs. A gate with n inputs has $2n$ timing arcs and therefore 2^{2n} variants (including the original cell). The number of devices in the cell can

[7]There are four timing arcs per non-unate input (e.g., select input of MUX).

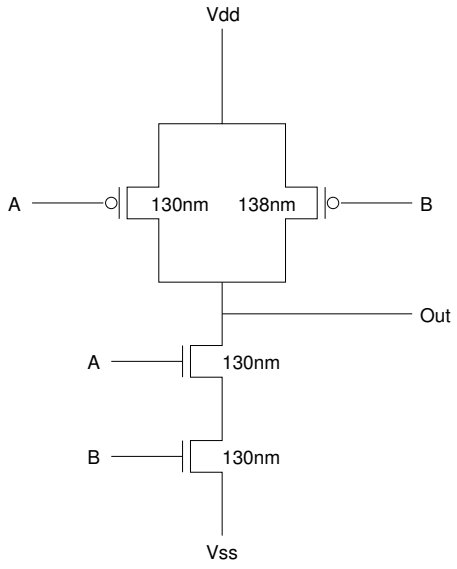


Figure IV.4: Gate length biasing of the transistors in NAND2X1 when only the rise and fall timing arcs from input A to the output are critical.

also limit the number of variants to 2^d, where d is the number of devices. Thus, the number of variants is: $min(2^{2n}, 2^d)$. Given a set of critical timing arcs, our goal is to assign biased L_{Gate} to some transistors in the cell and nominal L_{Gate} to the remaining transistors such that: (1) critical timing arcs have a delay penalty of under 1% with respect to the original unbiased cell, and (2) cell leakage power is minimized. Assignment of L_{Gate} to transistors in a cell, given a set of critical timing arcs, can be done by analyzing the cell topology for simple cells. However, we automate the process in the following manner. We enumerate all configurations for each cell in which nominal L_{Gate} is assigned to some transistors and biased L_{Gate} to the others. For each configuration we find the delay and leakage under a canonical output load of an inverter (INVX1) using SPICE. Now for each possible subset of timing arcs that can be simultaneously critical, one biasing configuration is chosen based on the two criteria given earlier. Figure IV.4 shows L_{Gate} biasing of the transistors in the simplest NAND cell (NAND2X1) when only the rise and fall timing arcs from input A to the output are critical. In this case only the PMOS device with B as its input can be slowed without penalizing the critical timing arcs.

IV.C.2 Optimization for Leakage

We use a sensitivity-based downsizing approach that is very similar to the one described in Section IV.B.2. We keep track of the slack on every timing arc and compute sensitivity for each timing arc. To limit the runtime and memory requirements, we first optimize at the cell level and then optimize at the transistor level for only the unbiased cells in the circuit.

IV.D Experiments and Results

We now describe our test flow for validation of the L_{Gate} biasing methodology, and present experimental results. Details of the testcases[8] used in our experiments are given in Table IV.3. The testcases are synthesized with the *Artisan TSMC* 130*nm library* using *Synopsys Design Compiler v2003.06-SP1* [11] with low-V_{th} cells only. To limit library characterization runtime, we restrict the library to variants of the following 25 most frequently used cells: CLKINVX1, INVX12, INVX1, INVX3, INVX4, INVX8, INVXL, MXI2X1, MXI2X4, NAND2X1, NOR2X1, NAND2X2, NAND2BX4, NAND2X4, NAND2X6, NAND2X8, NAND2XL, NOR2X2, NOR2X4, NOR2X6, NOR2X8, OAI21X4, XNOR2X1, XNOR2X4, XOR2X4. To identify the most frequently used cells, we synthesize our testcases with the complete library and select the 25 most frequently used cells. The delay constraint is kept tight so that the post-synthesis delay is close to minimum achievable delay. Since only low-V_{th} cells are used in synthesis, leakage of our testcases is very high and generally more than dynamic power.

We consider up to two gate lengths and two threshold voltages. We perform experiments for the following scenarios: (1) Single-V_{th}, single-L_{Gate} (SVT-SGL), (2) Dual-V_{th}, single L_{Gate} (DVT-SGL), (3) Single-V_{th}, dual-L_{Gate} (SVT-DGL), and (4) Dual-V_{th}, dual L_{Gate} (DVT-DGL). The dual-V_{th} flow uses nominal and low values of V_{th} while the single-V_{th} flow uses only the low value of V_{th}.

[8]To handle sequential testcases, we convert them to combinational circuits by treating all flip-flops as primary inputs and primary outputs.

Table IV.3: Testcases used in our experiments and their details. All cells in each circuit are low-V_{th} cells and dynamic power is calculated assuming an activity factor of 0.02. We use typical corner (typical process, $1.2V$, 25^oC) for delay and power analysis.

Circuit	Source	#Cells	Delay (ns)	Leakage (mW)	Dynamic (mW)
s9234	ISCAS'89	861	0.437	0.7074	0.3907
c5315	ISCAS'85	1442	0.556	1.4413	1.5345
c7552	ISCAS'85	1902	0.485	1.8328	2.0813
s13207	ISCAS'89	1957	0.904	1.3934	0.6296
c6288	ISCAS'85	4289	2.118	3.5994	8.0316
alu128	Opencores.org[8]	7536	2.306	5.1571	4.4177
s38417	ISCAS'89	7826	0.692	4.9381	4.2069

STMicroelectronics $130nm$ device models are used with the two V_{th} values each for PMOS and NMOS transistors (PMOS: -0.09V and -0.17V; NMOS: 0.11V and 0.19V). We use *Cadence SignalStorm v4.1* [4] (with *Synopsys HSPICE* [13]) for delay and power characterization of cell variants. *Synopsys Design Compiler* [11] is used to measure circuit delay, dynamic power and leakage power. We assume an activity factor of 0.02 for dynamic power calculation in all our experiments. We do not assume any wireload models or wire parasitics, as a result of which the dynamic power and delay overheads of L_{Gate} biasing are conservative (i.e., overestimated). Delay and power analyses are performed at the typical corner (typical process, $1.2V$, 25^oC). All experiments are run on an Intel Xeon 1.4GHz computer with 2GB of RAM.

IV.D.1 Leakage Reduction

Table IV.4 shows the leakage savings and delay penalties due to L_{Gate} biasing for each cell in our library. The results strongly support our hypothesis that

Table IV.4: Leakage reduction and delay penalty due to gate length biasing for all 25 cells in our library.

Cell	Low V_{th}		Nominal V_{th}	
	Leakage Reduction (%)	Delay Penalty (%)	Leakage Reduction (%)	Delay Penalty (%)
CLKINVX1	30.02	5.59	34.12	5.54
INVX12	30.28	4.70	36.27	6.87
INVX1	29.45	5.08	33.63	5.12
INVX3	30.72	5.68	35.67	5.52
INVX4	30.01	5.36	35.38	6.28
INVX8	29.97	6.75	35.73	5.25
INVXL	24.16	4.91	28.05	4.79
MXI2X1	23.61	5.45	27.26	5.97
MXI2X4	27.77	6.28	33.27	6.76
NAND2BX4	29.86	7.70	34.07	7.52
NAND2X1	33.19	5.32	37.03	5.58
NAND2X2	32.55	6.13	36.64	6.47
NAND2X4	32.21	6.54	36.95	6.63
NAND2X6	31.76	11.37	37.09	6.75
NAND2X8	31.70	6.07	37.14	7.29
NAND2XL	28.81	5.39	29.86	5.50
NOR2X1	27.42	5.47	32.58	5.39
NOR2X2	28.54	5.92	34.06	5.66
NOR2X4	28.85	6.61	34.25	8.21
NOR2X6	28.78	7.29	34.18	7.47
NOR2X8	28.76	6.51	34.40	6.96
OAI21X4	32.89	6.98	37.63	6.82
XNOR2X1	28.22	5.75	33.06	7.59
XNOR2X4	30.96	4.86	37.99	7.76
XOR2X4	30.87	7.92	37.98	6.85

small L_{Gate} biases can afford significant leakage savings. We now use our leakage optimization approach to selectively bias cells on non-critical paths. Table IV.5 shows the leakage reduction, and Table IV.6 shows the dynamic power penalty and total power reduction for our testcases when L_{Gate} biasing is applied without dual-V_{th} assignment. Tables IV.7 and IV.8 shows results when L_{Gate} biasing is applied together with the dual-V_{th} approach. To show the effectiveness of L_{Gate} biasing with loose delay constraints, results when the delay constraint is relaxed are also shown for each circuit. The leakage reductions primarily depend on the slack profile of the circuit. If many paths have near-zero slacks, then the leakage reductions are smaller. As the delay penalty increases, more slack is available on paths and larger leakage reductions are seen. We observe that leakage reductions are smaller when the circuit has already been optimized using dual-V_{th} assignment. This is expected, because dual-V_{th} assignment consumes slack on non-critical paths, reducing the slack available for L_{Gate} optimization. We also observe larger leakage reductions in sequential circuits. This is because circuit delay is determined by the slowest of the many combinational stages in sequential circuits, and the percentage of non-critical paths is typically higher in sequential circuits.

Our leakage models do not include gate leakage, which can marginally increase due to biasing. Gate leakage is composed of gate length-dependent (gate-to-channel (I_{gc}) and gate-to-body (I_{gb}) tunneling) and independent components (edge direct tunneling ($I_{gs}+I_{gd}$)). The gate length-independent component, which stems from the gate-drain and gate-source overlap regions, is not affected by biasing. To assess the change in gate length-dependent components due to biasing we perform SPICE simulations to report the gate-to-channel leakage[9] for nominal and biased devices. We use $90nm$ *BSIM4* device models from a leading foundry that model all five components of gate leakage described in *BSIM v4.4.0*. Table IV.9 shows the gate and subthreshold leakage for biased and unbiased nominal V_{th} NMOS and PMOS devices of $1\mu m$ width at 25^oC and 125^oC. The reductions in subthreshold and gate leakage, as well as the total leakage reduction, are shown. Based on these

[9]The gate-to-body component is two orders of magnitude smaller than the gate-to-channel component, and is therefore excluded from this analysis.

Table IV.5: Impact of gate length biasing on leakage for single threshold-voltage designs. Delay penalty constraint is set to 0%, 2.5%, and 5% for each of the testcases. (Note: Delay penalty for SVT-SGL is always set to 0% due to the non-availability of V_{th} and L_{Gate} knobs. SVT-DGL is slower than SVT-SGL for delay penalties of 2.5% and 5%.)

Test	Delay (ns)	SVT-SGL Leakage (mW)	SVT-DGL Leakage (mW)	Reduction Leakage (%)	CPU (s)
s9234	0.437	0.7074	0.5023	28.99	1.81
	0.447	0.7074	0.5003	29.28	1.79
	0.458	0.7074	0.4983	29.56	1.79
c5315	0.556	1.4413	1.2552	12.91	5.60
	0.570	1.4413	1.0415	27.74	5.80
	0.584	1.4413	1.0242	28.94	5.79
c7552	0.485	1.8328	1.4447	21.18	10.97
	0.497	1.8328	1.3665	25.44	11.08
	0.509	1.8328	1.3177	28.10	10.89
s13207	0.904	1.3934	0.9845	29.35	11.46
	0.927	1.3934	0.9778	29.83	11.31
	0.949	1.3934	0.9758	29.97	11.27
c6288	2.118	3.5994	3.3391	7.23	70.51
	2.171	3.5994	2.8461	20.93	74.79
	2.224	3.5994	2.7415	23.83	70.11
alu128	2.306	5.1571	4.5051	12.64	270.00
	2.363	5.1571	3.5992	30.21	212.97
	2.421	5.1571	3.5900	30.39	211.47
s38417	0.692	4.9381	3.4847	29.43	225.18
	0.710	4.9381	3.4744	29.64	225.68
	0.727	4.9381	3.4713	29.70	221.35

Table IV.6: Impact of gate length biasing on dynamic and total power (assuming an activity factor of 0.02) for single threshold-voltage designs. Delay penalty constraint is set to 0%, 2.5%, and 5% for each of the testcases.

Test	Delay	SVT-SGL		SVT-DGL		Reduction	
	(ns)	Dynamic (mW)	Total (mW)	Dynamic (mW)	Total (mW)	Dynamic (%)	Total (%)
s9234	0.437	0.3907	1.0981	0.4005	0.9028	-2.50	17.79
	0.447	0.3907	1.0981	0.4006	0.9008	-2.52	17.96
	0.458	0.3907	1.0981	0.4006	0.8988	-2.51	18.15
c5315	0.556	1.5345	2.9758	1.5455	2.8007	-0.72	5.88
	0.570	1.5345	2.9758	1.5585	2.6000	-1.56	12.63
	0.584	1.5345	2.9758	1.5604	2.5846	-1.69	13.15
c7552	0.485	2.0813	3.9141	2.0992	3.5439	-0.86	9.46
	0.497	2.0813	3.9141	2.1042	3.4707	-1.10	11.33
	0.509	2.0813	3.9141	2.1084	3.4261	-1.30	12.47
s13207	0.904	0.6296	2.0230	0.6448	1.6293	-2.42	19.46
	0.927	0.6296	2.0230	0.6449	1.6226	-2.42	19.79
	0.949	0.6296	2.0230	0.6446	1.6204	-2.39	19.90
c6288	2.118	8.0316	11.6310	8.0454	11.3845	-0.17	2.12
	2.171	8.0316	11.6310	8.0931	10.9392	-0.77	5.95
	2.224	8.0316	11.6310	8.1051	10.8466	-0.92	6.74
alu128	2.306	4.4177	9.5748	4.4429	8.9480	-0.57	6.55
	2.363	4.4177	9.5748	4.4818	8.0810	-1.45	15.60
	2.421	4.4177	9.5748	4.4826	8.0726	-1.47	15.69
s38417	0.692	4.2069	9.1450	4.2765	7.7612	-1.65	15.13
	0.710	4.2069	9.1450	4.2778	7.7522	-1.69	15.23
	0.727	4.2069	9.1450	4.2779	7.7492	-1.69	15.26

Table IV.7: Impact of gate length biasing on leakage for dual threshold-voltage designs. Delay penalty constraint is set to 0%, 2.5%, and 5% for each of the testcases.

Test	Delay (ns)	DVT-SGL Leakage (mW)	DVT-DGL Leakage (mW)	Reduction Leakage (%)	CPU (s)
s9234	0.437	0.0984	0.0722	26.60	1.86
	0.447	0.0914	0.0650	28.81	1.89
	0.458	0.0873	0.0609	30.20	1.83
c5315	0.556	0.3772	0.3391	10.11	5.74
	0.570	0.2871	0.2485	13.45	6.21
	0.584	0.2401	0.1986	17.27	6.14
c7552	0.485	0.6798	0.6655	2.10	10.40
	0.497	0.4698	0.4478	4.68	10.51
	0.509	0.3447	0.3184	7.63	10.55
s13207	0.904	0.1735	0.1247	28.09	11.59
	0.927	0.1561	0.1066	31.68	11.73
	0.949	0.1536	0.1027	33.14	11.76
c6288	2.118	1.9733	1.9517	1.09	79.25
	2.171	1.2258	1.1880	3.08	79.25
	2.224	0.8446	0.8204	2.87	77.28
alu128	2.306	0.6457	0.5184	19.73	240.09
	2.363	0.6151	0.4970	19.21	262.37
	2.421	0.5965	0.4497	24.62	277.99
s38417	0.692	0.5862	0.4838	17.46	238.62
	0.710	0.5637	0.4189	25.69	238.99
	0.727	0.5504	0.4067	26.11	234.94

Table IV.8: Impact of gate length biasing on dynamic and total power (assuming an activity factor of 0.02) for dual threshold-voltage designs. Delay penalty constraint is set to 0%, 2.5%, and 5% for each of the testcases.

Test	Delay	DVT-SGL		DVT-DGL		Reduction	
	(ns)	Dynamic (mW)	Total (mW)	Dynamic (mW)	Total (mW)	Dynamic (%)	Total (%)
s9234	0.437	0.3697	0.4681	0.3801	0.4523	-2.81	3.37
	0.447	0.3691	0.4604	0.3798	0.4448	-2.90	3.39
	0.458	0.3676	0.4549	0.3784	0.4393	-2.95	3.41
c5315	0.556	1.4298	1.8070	1.4483	1.7874	-1.29	1.09
	0.570	1.4193	1.7064	1.4390	1.6875	-1.39	1.11
	0.584	1.4119	1.6520	1.4328	1.6314	-1.48	1.24
c7552	0.485	1.9332	2.6130	1.9393	2.6048	-0.32	0.31
	0.497	1.9114	2.3812	1.9210	2.3689	-0.50	0.52
	0.509	1.8994	2.2441	1.9107	2.2291	-0.59	0.67
s13207	0.904	0.5930	0.7664	0.6069	0.7316	-2.35	4.54
	0.927	0.5920	0.7481	0.6060	0.7127	-2.37	4.73
	0.949	0.5919	0.7455	0.6060	0.7087	-2.39	4.93
c6288	2.118	7.7472	9.7205	7.7572	9.7089	-0.13	0.12
	2.171	7.5399	8.7657	7.5574	8.7454	-0.23	0.23
	2.224	7.4160	8.2606	7.4283	8.2487	-0.17	0.14
alu128	2.306	3.9890	4.6347	4.0353	4.5537	-1.16	1.75
	2.363	3.9837	4.5988	4.0242	4.5212	-1.02	1.69
	2.421	3.9817	4.5782	4.0378	4.4875	-1.41	1.98
s38417	0.692	3.8324	4.4186	3.8680	4.3518	-0.93	1.51
	0.710	3.8309	4.3946	3.8861	4.3050	-1.44	2.04
	0.727	3.8306	4.3810	3.8849	4.2916	-1.42	2.04

results, we conclude that the increase in gate leakage due to biasing is negligible. Furthermore, since biasing is a runtime leakage reduction approach and reduces the leakage of switching devices, the operating temperature is likely to be higher than room temperature – in this scenario gate leakage is not a major portion of total leakage. When the operating temperature is elevated, the reduction in total leakage is approximately equal to the reduction in subthreshold leakage, and total leakage reductions similar to the results presented in Tables IV.5 and IV.7 are expected.[10] Gate leakage is predicted to increase with technology scaling; technologies under $65nm$, however, are likely to adopt high-k gate dielectrics which will tremendously reduce gate leakage, so in terms of scalability, subthreshold leakage remains the key problem at high operating temperatures. We also note that because the vertical electric fields do not increase due to biasing, negative-bias thermal instability (NBTI) is not expected to increase with biasing [155].

IV.D.2 Manufacturability and Process Effects

We now investigate the manufacturability and process variability implications of our L_{Gate} biasing approach. As our method relies on the biasing of drawn gate length, it is important to correlate this with the actual printed gate length on the wafer. This is even more important as the bias we introduce in gate length is of the same order as the typical CD tolerances in manufacturing processes. Moreover, we expect larger gate lengths to have better printability properties leading to less CD, and hence leakage, variability. To validate our multiple gate length approach in a post-manufacturing setup, we follow a RET and process simulation flow for an example cell master.

We use the layout of a generic AND2X6 cell and perform model-based OPC on it using *Calibre v9.3_2.5* [6].[11] The printed image of the cell is then calculated using *dense* simulation in Calibre. The layout of the cell along with

[10] We report subthreshold leakage at 25^oC. Although the subthreshold leakage itself increases significantly with temperature, the percentage reduction in it due to biasing does not change much.

[11] Model-based OPC is performed using annular optical illumination with $\lambda = 248nm$, $NA = 0.7$, $\sigma_1 = 0.85$ and $\sigma_2 = 0.35$.

Table IV.9: Impact of gate length biasing on subthreshold leakage and gate tunneling leakage of $90nm$ PMOS and NMOS devices of $1\mu m$ width at different temperatures. Total leakage reductions are high even when gate leakage is considered.

Device	Temp (^oC)	Subthreshold Leakage (nW)		
		Unbiased	Biased	Reduction
PMOS	25	6.45	4.21	34.73%
NMOS	25	12.68	8.43	33.52%
PMOS	125	116.80	79.91	31.58%
NMOS	125	115.90	83.58	27.89%
Device	Temp (^oC)	Gate Tunneling Leakage (nW)		
		Unbiased	Biased	Reduction
PMOS	25	2.01	2.03	-1.00%
NMOS	25	6.24	6.25	-0.16%
PMOS	125	2.17	2.20	-1.38%
NMOS	125	6.62	6.69	-1.05%
Device	Temp (^oC)	Total Leakage (nW)		
		Unbiased	Biased	Reduction
PMOS	25	8.46	6.24	26.24%
NMOS	25	18.92	14.68	22.41%
PMOS	125	118.97	82.11	30.98%
NMOS	125	122.52	90.27	26.32%

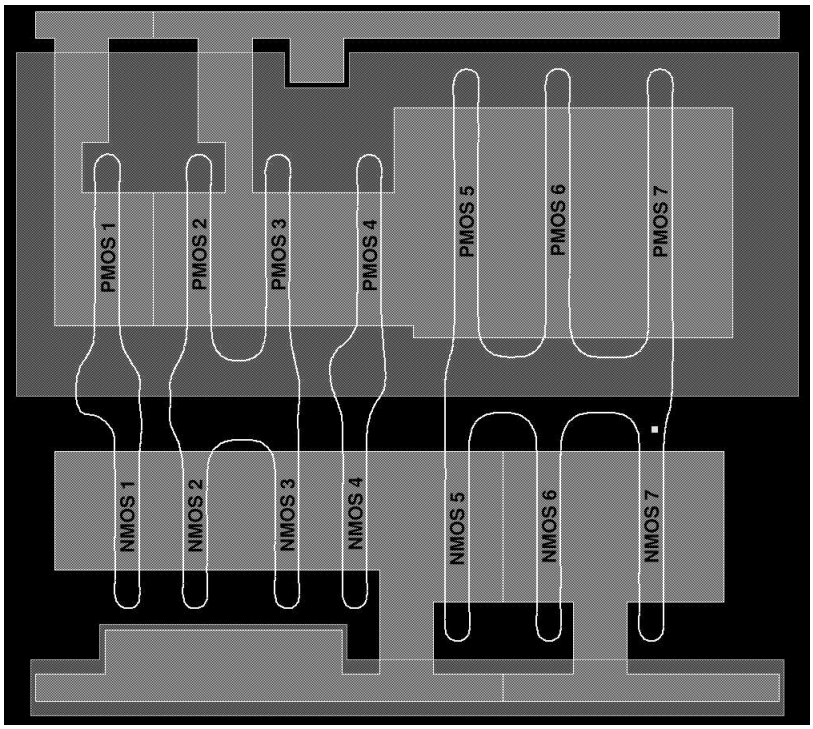


Figure IV.5: Layout of a generic AND2X6 cell with simulated printed gate lengths.

printed gate lengths of all devices in it is shown in Figure IV.5. We measure the L_{Gate} for every device in the cell, for both biased and unbiased versions. The printed gate lengths for the seven NMOS and seven PMOS devices labeled in Figure IV.5 are shown in Table IV.10. As expected, biased and unbiased gate lengths track each other well. There are some outliers that may be due to the relative simplicity of the OPC model being used. High correlation between *printed* dimensions of biased and unbiased versions of the cells shows that the benefits of biasing estimates using *drawn* dimensions will not be lost after RET application and the manufacturing process.

Another potentially valuable benefit of slightly larger gate lengths is the possibility of improved printability. Minimum poly spacing is larger than poly gate length, so that the process window (which is constrained by the minimum resolvable dimension) tends to be larger as gate length increases, even though poly spacing decreases. For example, the exposure latitude values for various depth

Table IV.10: Comparison of printed dimensions of unbiased and biased versions of AND2X6. The unbiased nominal gate length is $130nm$ while the biased nominal is $138nm$. Note the high correlation between unbiased and biased versions.

Device Number	Gate Length (nm)					
	PMOS			NMOS		
	Unbiased	Biased	Diff.	Unbiased	Biased	Diff.
1	128	135	+7	129	135	+6
2	127	131	+4	126	131	+5
3	127	131	+4	127	131	+4
4	124	131	+7	126	133	+7
5	124	131	+7	124	132	+8
6	124	132	+8	124	132	+8
7	127	135	+8	127	135	+8

of focus values with the same illumination system for $130nm$ and $138nm$ lines is shown in Table IV.11.[12]

[12]The process simulation was performed using *Prolith v8.1.2* [10].

Table IV.11: Process window improvement with gate length biasing. The CD tolerance is kept at $13nm$. ELAT = Exposure latitude.

Defocus (μm)	ELAT (%) for $130nm$	ELAT (%) for $138nm$
-0.2	4.93	5.30
0.0	6.75	7.26
0.2	5.69	6.24

IV.D.3 Process Variability

A number of sources of variation can cause fluctuations in gate length, and hence in performance and leakage. This has been a subject of much discussion in the recent literature (e.g., [150, 36]). Up to 20× variation in leakage has been reported in production microprocessors [29]. For leakage, the reduction in variation post-biasing is likely to be substantial as the larger gate length is closer to the "flatter" region of the V_{th} vs. L_{Gate} curve. To validate this intuition, we study the impact of gate length variation on leakage and performance, both pre- and post-biasing, using a simple worst-case approach. We assume the CD variation budget to be $\pm 10nm$. The performance and leakage of the testcase circuits is measured at the worst-case, nominal and best-case process corners which consider just gate length variation. Best-case (worst-case) refers to the process corner at which a metric is at its most (least) desirable value. Thus, best-case for delay corresponds to small gate lengths, while best-case for leakage corresponds to having large gate lengths. We report the results for the DVT-DGL approach in which biasing is performed along with dual-V_{th} assignment in Table IV.12. For the seven testcases, we see up to a 41% reduction in leakage power uncertainty caused by linewidth variation. Such large reductions in uncertainty make biasing a very compelling leakage reduction technique. The impact on delay variability reduction is negligible. Leakage variability reduction depends on the number of cells, especially low-V_{th} cells, that get biased. We note that the corner-case analysis only models the inter-die component of variation.

To assess the impact of both within-die (WID) and die-to-die (DTD) components of variation, we run 10,000 Monte-Carlo simulations with $\sigma_{WID} = \sigma_{DTD} = 3.33nm$. The variations are assumed to follow a Gaussian distribution with no correlations. We compare the results for three dual-V_{th} scenarios: unbiased (DVT-SGL), biased (DVT-DGL) and uniformly biased (when gate lengths of all transistors in the design are biased by $8nm$). Leakage distributions for the testcase $alu128$ are shown in Figure IV.6. Note that in uniform biasing all devices are biased and the circuit delay no longer satisfies timing constraints.

Table IV.12: Reduction in performance and leakage power uncertainty with biased gate length in presence of inter-die variations. The uncertainty spread is specified as a percentage of nominal. The results are given for dual-V_{th} and the biasing is $8nm$. *BC* corresponds to best-case, *WC* to worst-case, and *NOM* to nominal analyses.

Circuit	Circuit Delay (ns)						
	Unbiased (DVT-SGL)			Biased (DVT-DGL)			% Spread
	WC	BC	NOM	WC	BC	NOM	Reduction
s9234	0.504	0.385	0.436	0.506	0.387	0.436	-0.53
c5315	0.642	0.499	0.556	0.643	0.501	0.556	0.71
c7552	0.559	0.433	0.485	0.559	0.433	0.485	0.46
s13207	1.029	0.797	0.904	1.031	0.800	0.904	0.35
c6288	2.411	1.888	2.118	2.411	1.889	2.118	0.13
alu128	2.631	2.045	2.305	2.640	2.053	2.306	-0.10
s38417	0.793	0.615	0.692	0.793	0.616	0.692	0.03
Circuit	Leakage (mW)						
	Unbiased (DVT-SGL)			Biased (DVT-DGL)			% Spread
	BC	WC	NOM	BC	WC	NOM	Reduction
s9234	0.0591	0.1898	0.0984	0.0467	0.1268	0.0722	38.76
c5315	0.2358	0.6883	0.3772	0.2176	0.5960	0.3391	16.38
c7552	0.4291	1.2171	0.6798	0.4226	1.1825	0.6655	3.57
s13207	0.1036	0.3401	0.1735	0.0807	0.2211	0.1247	40.65
c6288	1.2477	3.5081	1.9733	1.2373	3.4559	1.9517	1.85
alu128	0.3827	1.2858	0.6457	0.3229	0.9641	0.5184	29.00
s38417	0.3526	1.1453	0.5862	0.3038	0.8966	0.4838	25.22

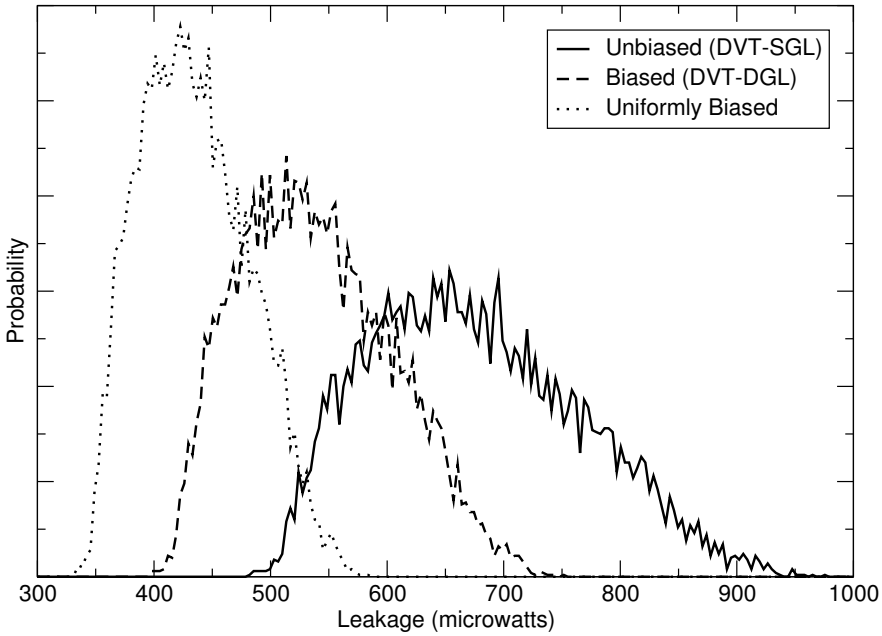


Figure IV.6: Leakage distributions for unbiased, uniformly biased, and cell-level selectively-biased alu128. Note the "left-shift" of the distribution with the introduction of biased devices in the design.

IV.D.4 Leakage Reduction from Transistor-Level Biasing

Table IV.13 presents the leakage power reductions from TLLB over CLLB. We see up to a 10% reduction in leakage power over CLLB. Since TLLB only biases devices of unbiased cells, circuits in which a large number of cells remain unbiased after CLLB are more amenable to TLLB. The leakage savings from TLLB come at the cost of increased library size. As described in Section IV.C.1, the library is composed of all $min(2^{2n}, 2^d)$ variants of each n-input cell. For the 25 cells, our library for TLLB was composed of a total of 920 variants. From the small leakage savings at the cost of significantly increased library size, we conclude that TLLB should only be performed for single- and double- input cells that are frequently used.

Table IV.13: Leakage power from transistor-level gate length biasing.

Circuit	Delay	Leakage			CPU (s)	
		CLLB	TLLB	Reduction	CLLB	TLLB
	(ns)	(mW)	(mW)	(%)	(s)	(s)
s9234	0.437	0.0722	0.0712	1.41	1.86	2.75
	0.447	0.0650	0.0628	3.39	1.89	2.38
	0.458	0.0609	0.0596	2.28	1.83	2.31
c5315	0.556	0.3391	0.3359	0.95	5.74	14.99
	0.570	0.2485	0.2368	4.71	6.21	15.29
	0.584	0.1986	0.1918	3.42	6.14	13.44
c7552	0.485	0.6655	0.6356	4.49	10.40	43.79
	0.497	0.4478	0.4438	0.89	10.51	43.22
	0.509	0.3184	0.2993	6.02	10.55	38.90
s13207	0.904	0.1247	0.1228	1.58	11.59	17.15
	0.927	0.1066	0.1055	1.08	11.73	15.62
	0.949	0.1027	0.1021	0.61	11.76	14.28
c6288	2.118	1.9517	1.9157	1.84	79.25	305.09
	2.171	1.1880	1.1555	2.74	79.46	289.56
	2.224	0.8203	0.8203	0.00	77.28	291.44
alu128	2.306	0.5184	0.4857	6.31	240.09	544.75
	2.363	0.4970	0.4492	9.62	262.37	609.13
	2.421	0.4497	0.4184	6.95	277.99	534.68
s38417	0.692	0.4838	0.4467	7.67	238.62	746.79
	0.710	0.4189	0.3982	4.93	238.99	507.62
	0.727	0.4067	0.3765	7.42	234.94	525.06

IV.E Impact of Biasing on Threshold Voltage Selection

In this section we:

- study the simultaneous use of V_{th} assignment and gate length biasing. Specifically, we empirically assess the impact of availability of gate length biasing on V_{th} selection, a decision that is typically made by the foundries.
- evaluate the effectiveness of foundry V_{th}'s on leakage improvements of large designs by comparing against different synthesized V_{th}'s.
- analyze the impact of leakage improvement obtained due to gate length biasing on top of different V_{th}'s.

Multiple foundry V_{th}'s increase manufacturing cost and impact wafer yield due to increase in number of masks and processing steps respectively. We show that a combination of gate length biasing and dual-V_{th} assignment provides cost-effective leakage reduction comparable to that of triple V_{th} assignment.

IV.E.1 Simultaneous Threshold Voltage Assignment and Biasing

In this section we compare the techniques of gate length biasing and V_{th} assignment, present the advantages of their simultaneous use, and motivate the need for judicious selection of the different threshold voltages that is aware of the availability of gate length biasing.

Multi-V_{th} (achieved through multiple doping concentrations) and gate length biasing both alter V_{th} to trade off leakage power against delay. Both techniques can be applied post-layout when accurate timing information is available, and do not require iterations with the synthesis, placement, and routing loop. However, there are certain differences between the two techniques. In multi-V_{th}, different threshold voltages are attained by changing the doping concentration, and

extra mask (NRE) and process (recurring) costs are involved. Gate length biasing, on the other hand, exploits SCE by increasing gate length to increase threshold voltage, and does not increase manufacturing cost.

Gate length biasing has several disadvantages in comparison to multi-V_{th}. Multi-V_{th} achieves a more favorable tradeoff between leakage and delay than biasing. Also, biasing increases input gate capacitance marginally, which can affect the delay of fanin cells due to increased loading. Increased gate capacitance also increases dynamic power, making biasing usable only when activity factors are small (< 0.1). Due to these shortcomings, gate length biasing cannot be used as a replacement for higher-cost multi-V_{th}.

Advantages of gate length biasing, in addition to lower process costs, include finer control over the leakage-delay tradeoff and significant reduction in leakage variability. Figure IV.7 shows the tradeoff between delay and I_{off} as the gate length is increased up to 10% for three foundry-set V_{th}. This finer control allows circuit-level leakage optimizers to reduce leakage by more than what is possible with the coarse control provided by V_{th} assignment only. Essentially, a simple optimizer can exploit the residual timing slacks after V_{th} assignment using gate length biasing to reduce leakage; a sophisticated optimizer, that simultaneously performs V_{th} assignment and biasing, will of course reduce leakage more effectively. Additionally, as discussed in Section IV.D.3, gate length biasing reduces leakage variability substantially. Due to the improvement in leakage and leakage variability provided by gate length biasing at no additional manufacturing cost, we propose to use gate length biasing with V_{th} assignment.

Table IV.14 presents a comparison of leakage reductions obtained on our testcases (details in Section IV.E.3) obtained with (a) two foundry-set V_{th}'s, LVT and SVT, (b) two foundry-set V_{th}'s, LVT and HVT, (c) three foundry-set V_{th}'s, LVT, SVT and HVT, (d) LVT and SVT with biasing and (e) LVT and HVT with biasing. We can observe that foundry dual-V_{th} combined with biasing achieves reductions comparable to foundry triple-V_{th}.

We now study the impact of availability of gate length biasing on V_{th} selection. Typically foundries select V_{th}'s for each process such that high leakage

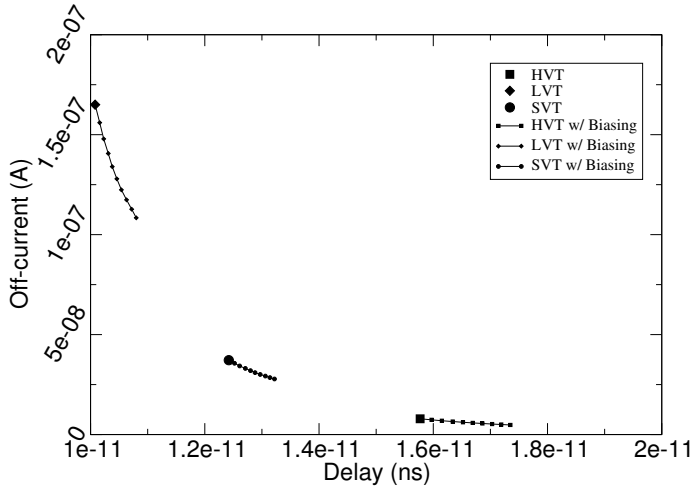


Figure IV.7: Off-current and delay of an NMOS device as its *VTH0* is modified for HVT, SVT, and LVT.

Table IV.14: Leakage reduction with: (a) two foundry-set V_{th}'s, LVT and SVT; (b) two foundry-set V_{th}'s, LVT and HVT; (c) three foundry-set V_{th}'s, LVT, SVT and HVT; (d) V_{th}'s, LVT and SVT with biasing; and (e) V_{th}'s, LVT and HVT with biasing.

Circuit	Leakage Reduction (%)				
	LVT + SVT	LVT + HVT	LVT + SVT + HVT	LVT + SVT + Biasing	LVT + HVT + Biasing
AES	65.89	64.03	74.41	70.15	67.94
OR1200	59.86	70.83	71.46	65.89	73.04
DES3	70.07	72.43	80.28	74.41	75.55

reductions are achieved for all designs that use them. To achieve high leakage reductions: (1) a large number of cells must be convertible to higher V_{th}'s, and (2) per-cell leakage reduction due to higher V_{th} assignment should be large. In real-world designs, logic blocks undergo many optimization steps and can have a large number of critical paths. Thus higher V_{th}'s must have a small delay penalty to allow a significant number of cells to be converted to higher V_{th}. Unfortunately, small delay penalty yields small per-cell leakage reduction on higher V_{th} assignment. Increasing the higher V_{th} increases the per-cell leakage reduction, but decreases the number of cells that can be converted to higher V_{th}. This occurs because of the larger delay penalty associated with the increased higher V_{th}. Similarly, lowering the higher V_{th} would allow more cells to get higher V_{th} assigned but the per-cell leakage reduction would decrease. Therefore, selection of a good set of V_{th}'s depends on the leakage-delay tradeoff due to V_{th}, as well as the slack distribution and structure of the design's netlist. Assessing the impact of availability of biasing on V_{th} assignment is complex and difficult, if not impossible, to generalize for different testcases. We empirically study the effect on real-world testcases and industry SPICE models.

IV.E.2 Threshold Voltage Customization

We use TSMC $100nm$ process that has three foundry-set V_{th}'s. Our experiments require more V_{th}'s because we study the impact of changing the available set of V_{th}'s on leakage reduction. We artificially generate V_{th}'s by modifying the *VTH0* parameter in the SVT SPICE device model. To test the accuracy of this method, we modify the *VTH0* of (foundry-set) SVT gradually until I_{off} and I_{on} (or delay) characteristics are similar to those of (foundry-set) HVT and LVT. Figures IV.8 and IV.9 plot the delay and I_{off} of the NMOS and PMOS devices in an inverter cell (*INVX4*) respectively. The NMOS and PMOS widths are $1.16\mu m$ and $1.64\mu m$ respectively and the inverter is loaded with $5fF$ capacitance. The figures also show the I_{off} and delay tradeoffs when *VTH0* is changed for HVT and LVT SPICE models. As can be seen, we are able to match the delay and I_{off} of HVT

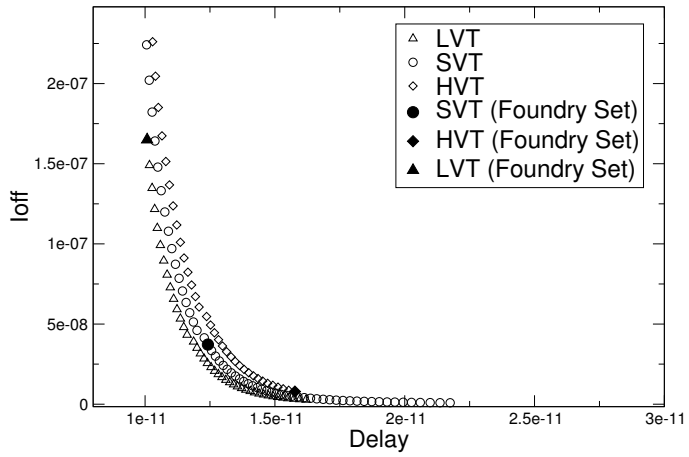


Figure IV.8: Off-current and delay of the NMOS device in INVX4 as *VTH0* is modified for HVT, SVT, and LVT.

and LVT very accurately by changing *VTH0* of SVT.

Increasing the number of available V_{th}'s improves the granularity of selection of best V_{th}'s but is time-consuming since characterization, a computationally intensive process, must be performed for all available V_{th}'s for each cell in our library. Therefore we strike a middle ground and increase the number of V_{th}'s from three to seven. We create four more V_{th}'s in addition to the foundry-set V_{th}'s, such that all V_{th} values are approximately equally spaced. Table IV.15 gives threshold voltages, I_{off}, and delay values for different V_{th}'s used in our study.

IV.E.3 Experiments and Results

We now present our experimental setup, results, and their assessment.

Experimental Setup

We alter the *VTH0* in *TSMC* 100*nm* SPICE device models (as explained in Section IV.E.2) to get SPICE models for our seven V_{th}'s. Library characterization is performed using *Cadence SignalStorm v04.10* [4] and *Synopsys HSPICE* [13] on *Artisan TSMC* standard-cell SPICE netlists for 50 combinational cells. We do not create cell variants for different V_{th}'s and gate biases for the 13 sequential

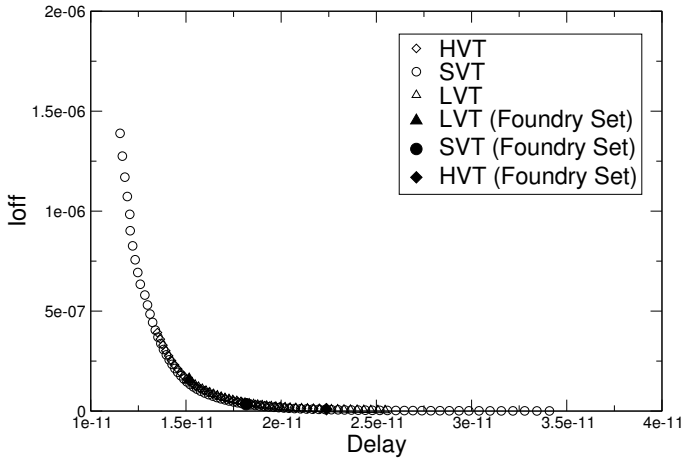


Figure IV.9: Off-current and delay of the PMOS device in INVX4 as *VTH0* is modified for HVT, SVT, and LVT.

Table IV.15: The seven threshold voltages used in our experiments.

Name	Description	NMOS (W=1.16μm)			PMOS (W=1.64μm)		
		V_{th} (V)	I_{off} (nA)	Delay (ps)	V_{th} (V)	I_{off} (nA)	Delay (ps)
HHVT	Ultra-high V_{th} (V_{th} of HVT is midway between HHVT and HSVT)	0.437	3.7	16.37	-0.330	5.1	23.69
HVT	Foundry-set high-V_{th}	0.402	7.5	14.86	-0.300	9.4	21.65
HSVT	V_{th} midway between HVT and SVT	0.362	17.6	13.48	-0.265	18.9	19.62
SVT	Foundry-set standard-V_{th}	0.327	37.2	12.42	-0.235	34.2	18.16
SLVT	V_{th} midway between SVT and LVT	0.292	78.5	11.31	-0.195	74.8	16.41
LVT	Foundry set low-V_{th}	0.257	164.2	10.38	-0.155	160.6	14.89
LLVT	Ultra-low V_{th} (V_{th} of LVT is midway between SLVT and LLVT)	0.222	338.0	9.66	-0.115	337.5	13.68

Table IV.16: Testcases used in our experiments and their details. All testcases were sourced from opencores.org.

Testcase	#Cells	LLVT		LVT		SLVT	
		Delay (ns)	Leakage (mW)	Delay (ns)	Leakage (mW)	Delay (ns)	Leakage (mW)
AES	22,371	1.134	9.46	1.214	4.61	1.294	2.24
OR1200	44,237	2.860	24.01	2.960	13.08	3.110	7.69
DES3	85,878	1.081	36.31	1.106	18.08	1.160	9.08

cells in our library, to conserve characterization runtime. Therefore, sequential cells are not optimized for leakage in our experiments.

We use three large testcases available in the public domain [8]. Testcases AES is an encryption core that implements the Rijndael algorithm. OR1200 is a 32-bit scalar RISC processor with five stage integer pipeline and on-chip SRAM. DES3 is a a combined encryption and decryption design that implements the triple data encryption standard algorithm. To obtain tight slack distributions for our testcases, we iteratively perform synthesis with *Synopsys DesignCompiler v2004.12* [11], each time decreasing the target clock cycle time by a small step, until a tight slack distribution is attained. Details about our testcases are summarized in Table IV.16. As can be seen, synthesis with LLVT (LVT) library reduces the clock cycle time but dramatically increases the leakage in comparison to the LVT (SLVT) library.

Our leakage reduction results are obtained from an industry leakage optimizer that is tuned for leakage reduction using V_{th} assignment and gate length biasing. It can optimize multi-million gate designs with high quality of results in tractable runtime. In all our experiments, we do not allow the clock cycle time to increase during optimization.

Experiments

We perform the following three analyses:

1. Comparison of leakage reductions from triple-V_{th} assignment, and dual-V_{th} assignment with biasing. I.e., we explore whether dual-V_{th} with biasing can be used as a replacement of triple-V_{th}.

2. Assessment of foundry-selected V_{th}'s by changing foundry-set V_{th}'s and comparing leakage reductions.

3. Assessment of the impact of availability of gate length biasing on the selection of V_{th}'s.

Leakage reductions from dual-V_{th} with biasing are compared with triple-V_{th} in Table IV.14. The clock cycle time is constant over all optimization runs, and bias values of $4nm$, $6nm$, $8nm$, and $10nm$ are used. We observe that triple-V_{th} reduces leakage with respect to dual-V_{th} (without biasing) by $\sim 8\%$ on average. However, when biasing is available, dual-V_{th} can yield reductions comparable to triple-V_{th} that involves higher process costs.

We now assess the choice of foundry-selected V_{th}'s when biasing is not available. We perform dual-V_{th} assignment with no increase in clock cycle time. The high V_{th} and low V_{th} are changed and leakage is measured. Table IV.17 presents the results for our testcases. For AES and DES3, HSVT yields the best reduction while for OR1200, HHVT is the best. In OR1200 a smaller percentage of paths is critical (i.e., the slack distribution is relatively loose) and this causes many cells to be assigned the highest V_{th}. HHVT has the highest leakage reduction per cell and consequently yields the best leakage reduction. Therefore, foundry-set V_{th}'s may not yield the best leakage reduction, and the slack distribution, netlist structure, and leakage-delay tradeoff must be understood prior to V_{th} selection.

It is also clear from Table IV.17 that reducing the low V_{th} does not yield good leakage reductions. One could expect that using lower low-V_{th} would loosen the design and make it more amenable to optimization. For instance, a gate that drives several gates can be assigned lower V_{th} to decrease its delay so that the slack

Table IV.17: Post-optimization leakage (in mW) for two low V_{th}'s and different high V_{th}'s.

Circuit	Low V_{th}	High V_{th}			
		HHVT	HVT	HSVT	SVT
AES	LLVT	3.100	2.689	2.182	1.812
	LVT	1.785	1.658	1.529	1.572
OR1200	LLVT	4.149	4.264	4.649	5.557
	LVT	3.699	3.818	4.219	5.254
DES3	LLVT	8.732	7.224	5.906	5.902
	LVT	5.600	4.986	4.614	5.4112

thus gained could be used to reduce the leakage of the gates it drives. However, low-V_{th} cells have an extremely high leakage cost, and the leakage reduction from exploiting the gained slack is typically smaller than the leakage increase due to use of the low-V_{th} cell. Therefore, low-V_{th} selection should be governed by timing only, and the maximum V_{th} that allows the circuit to meet timing must be used as the low-V_{th}.

To assess the impact of availability of gate length biasing on V_{th} selection, we perform leakage optimization for three different low V_{th}'s for each of our testcases. The clock cycle time and absolute leakage values differ for the three low V_{th}'s as shown in Table IV.16. For each testcase and low V_{th} combination, we change the set of available gate length biases and run experiments to identify the optimum (i.e., yielding the greatest leakage reduction) high V_{th}. Table IV.18 shows the leakage reduction achieved for testcase AES synthesized under three low V_{th} for different available gate bias values and high V_{th}.

The following observations may be made from the results:

- Leakage reduction increases as more biases are allowed for optimization. However, the benefit progressively diminishes as the number of biases becomes large.

Table IV.18: Leakage reduction for different high-V_{th} with different maximum gate length biases for AES. Best leakage reductions are shown in **bold**.

Lower V_{th}	Max. bias	Higher V_{th}			
		HHVT	HVT	HSVT	SVT
LLVT	no biasing	47.94	51.37	56.78	**61.58**
	4nm	54.07	57.24	62.09	**64.88**
	6nm	54.97	57.90	62.84	**65.50**
	8nm	55.35	58.21	63.10	**65.79**
	10nm	55.87	58.49	63.5	**66.06**
LVT	no biasing	61.27	64.03	**66.83**	65.89
	4nm	64.89	66.68	68.23	**68.35**
	6nm	65.66	67.31	68.85	**69.19**
	8nm	65.97	67.70	69.12	**69.72**
	10nm	66.21	67.90	69.46	**70.15**
SLVT	no biasing	64.42	**65.33**	63.23	49.08
	4nm	67.10	**67.93**	66.43	54.10
	6nm	67.74	**68.72**	67.25	56.10
	8nm	68.02	**69.10**	67.87	57.55
	10nm	68.21	**69.41**	68.34	58.63

- As the low V_{th} is increased, the optimum higher V_{th} also increases. This indicates that V_{th}'s should be neither spaced wide apart nor placed too close, so that the leakage-delay tradeoff is effectively covered.

- Availability of gate length biasing has small impact on optimum high V_{th}. For LVT as the lower V_{th}, we observe that the optimum high V_{th} shifts from HSVT to SVT when biasing becomes available for the testcase AES. Similar trends were observed for DES3.

Table IV.19 shows the optimum high V_{th} for all three of our testcases.

Table IV.19: Best high V_{th} for three low V_{th}'s and maximum bias of $4nm$, $6nm$, $8nm$, and $10nm$. Corresponding leakage savings are also shown.

Circuit	Low V_{th}	No biasing		$4nm$		$4, 6nm$	
		Best High V_{th}	Saving (%)	Best High V_{th}	Saving (%)	Best High V_{th}	Saving (%)
AES	LLVT	SVT	61.58	SVT	64.88	SVT	65.50
	LVT	HSVT	66.83	SVT	68.36	SVT	69.20
	SLVT	HVT	65.33	HVT	67.93	HVT	68.72
OR1200	LLVT	HVT	82.24	HVT	83.26	HVT	83.62
	LVT	HHVT	71.74	HHVT	72.74	HHVT	73.02
	SLVT	HHVT	55.49	HHVT	57.98	HHVT	58.87
DES3	LLVT	SHVT	80.12	SVT	82.97	SVT	83.36
	LVT	SHVT	74.48	SHVT	76.65	SHVT	77.03
	SLVT	HHVT	66.14	HVT	69.57	HVT	70.10

Circuit	Low V_{th}	$4, 6, 8nm$		$4, 6, 8, 10nm$	
		Best High V_{th}	Saving (%)	Best High V_{th}	Saving (%)
AES	LLVT	SVT	65.79	SVT	66.03
	LVT	SVT	69.72	SVT	70.15
	SLVT	HVT	69.10	HVT	69.41
OR1200	LLVT	HVT	83.90	HVT	83.90
	LVT	HHVT	73.09	HHVT	73.52
	SLVT	HHVT	59.56	HHVT	60.14
DES3	LLVT	SVT	83.67	SVT	83.90
	LVT	SHVT	77.28	SHVT	77.49
	SLVT	HVT	70.42	HVT	70.61

IV.F Gate Length Biasing Using Lagrangian Relaxation

We presented a sensitivity-based downsizing approach to optimize leakage with gate length biasing in Section IV.B.2. The sensitivity-based approach is a greedy approach, and while fast and flexible, can yield suboptimal results. Several circuit optimization approaches frame the gate sizing problem as a geometric program [31]. Geometric programming refers to the class of optimization problems in which the objective and the constraints are posynomial functions. Geometric programs are not convex in their original form but become convex under an exponential transformation, and efficient optimization techniques can be used to solve them. Chen et al. [44] presented a Lagrangian relaxation-based gate and wire sizing approach. Boyd et al. [32] used an interior-point geometric program solver. Joshi et al. [89] used a customized method that is of the truncated pseudo-Newton type.

We alter the Lagrangian relaxation-based gate and wire sizing approach proposed in [44] for leakage optimization through gate length biasing. Lagrangian relaxation transfers some or all constraints to the objective function by multiplying them with constants known as Lagrange multipliers. The original problem is known as the primal problem and the transformed problem is the Lagrangian relaxation subproblem. For constrained convex problems, it can be shown that there exists a set of Lagrange multipliers under which the optimal solutions of the primal problem and the Lagrangian relaxation subproblem coincide. Thus, if the Lagrangian relaxation subproblem can be solved for the minimum cost, and the set of Lagrange multipliers that maximize the minimum cost of the subproblem can be found (known as the dual problem), then the optimal solution to the Lagrangian relaxation subproblem is the optimal solution to the primal problem. We now present our problem formulations customized to the leakage optimization through gate biasing problem. General details of the method are presented in [44].

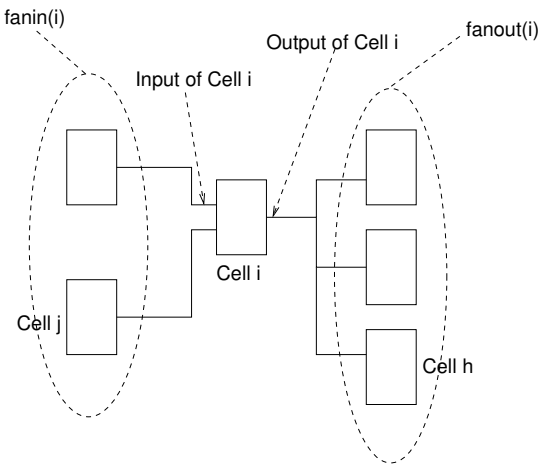


Figure IV.10: Cell i and its fanin and fanout cells.

IV.F.1 Nomenclature and Models

- There are n cells in the circuit.
- $\mathbf{x} \in R^n$ is the vector of biases assigned to all cells. x_i is the bias assigned to cell i. L_i and U_i represent the lower and upper bounds respectively on x_i.
- $P(\mathbf{x})$ represents the leakage *change* of the circuit when $\mathbf{x}$ is applied.
- $P(x_i)$ is the leakage change of cell i when bias x_i is applied. $P(\mathbf{x}) = \sum_{i=1}^{n} P(x_i)$.
- We assume leakage to be a quadratic function of the bias. $P(x_i) = \alpha_i x_i + \beta_i x_i^2$.
- The arrival and required times at the output of cell i are denoted by A_i and R_i respectively.
- Delay for each timing arc is modeled but rise and fall delays are set equal. Delay of cell i from its fanin cell j is represented by D_{ji}. Figure IV.10 illustrates the fanin and fanout cells. Wire delays are assumed to be zero.

$$D_{ji} = d_{ji} + r_{ji}\mathcal{C}_i$$

where d_{ji} is the intrinsic delay of the cell, r_{ji} is the drive resistance, and $\mathcal{C}_i$ is the loading capacitance.

- Intrinsic delay is a linear function of bias.

$$d_{ji} = \tilde{d}_{ji} + \hat{d}_{ji} x_i$$

where $\tilde{d}_{ji}$ is the nominal intrinsic delay (i.e., intrinsic delay of the nominal cell) and $\hat{d}_{ji}$ is the linear coefficient. Similarly, drive resistance is also a linear function of bias.

$$r_{ji} = \tilde{r}_{ji} + \hat{r}_{ji} x_i$$

- Input capacitance of cell i for the input pin connected to fanin cell j is represented by c_{ji}. It is a linear function of bias and is given by:

$$c_{ji} = \tilde{c}_{ji} + \hat{c}_{ji} x_i$$

Loading capacitance of cell i, $\mathcal{C}_i$ is given by:

$$\mathcal{C}_i = \sum_{h \in fanout(i)} c_{ih}$$

- The delay can be written as:

$$D_{ji} = \tilde{d}_{ji} + \hat{d}_{ji} x_i + (\tilde{r}_{ji} + \hat{r}_{ji} x_i) \times \sum_{h \in fanout(i)} \tilde{c}_{ih} + \hat{c}_{ih} x_h$$

where $fanout(i)$ is the set of all cells connected to the fanout of i.

To make the delay model a convex function, we neglect the $x_i x_h$ terms which are expected to be small. Our optimization uses the following delay model:

$$D_{ji} = \tilde{d}_{ji} + \hat{d}_{ji} x_i + \tilde{r}_{ji} \times \sum_{h \in fanout(i)} \tilde{c}_{ih} + \hat{c}_{ih} x_h + \hat{r}_{ji} x_i \times \sum_{h \in fanout(i)} \tilde{c}_{ih}$$

- We add a source node and a sink node to the circuit graph. The source is connected to all primary inputs and data pins of sequential cells; all primary outputs and Q pins are connected to the sink. Nodes in the circuit graph are indexed in reverse topological order with the source and sink nodes being numbered as $n+1$ and 0 respectively as in [44].

IV.F.2 Lagrangian Relaxation-Based Solution

Primal Problem (PP)

The primal problem is obtained by partitioning the constraints on path delays into constraints on delays of the gates. The primal problem is a geometric program and is given as follows.

- Minimize :

$$P(x) = \sum_{i=0}^{n+1} \alpha_i x_i + \beta_i x_i^2$$

- Subject to:

$$\begin{aligned} A_0 &\leq R_0 \\ 0 &\leq A_{n+1} \\ A_j + D_{ji} &\leq A_i \qquad i = 1, \ldots, n; j \in fanin(i) \\ L_i &\leq x_i \leq U_i \qquad i = 1, \ldots, n \end{aligned}$$

Lagrangian Relaxation Subproblem (LRS)

- Minimize :

$$\begin{aligned} Q(x) &= \sum_{i=0}^{n+1} \alpha_i x_i + \beta_i x_i^2 + \lambda_{0,0}(A_0 - R_0) + \lambda_{n+1,n+1}(-A_{n+1}) \\ &+ \sum_{i=1}^{n} \sum_{j \in fanin(i)} \lambda_{j,i}(A_j + D_{ji} - A_i) \end{aligned}$$

- Subject to :

$$L_i \leq x_i \leq U_i \quad i = 0, \ldots, n$$

LRS Simplification and Solution

As described in [44], we apply the Kuhn-Tucker conditions to derive optimality conditions on λ. Setting $\partial Q / \partial a_i = 0$, we get:

$$\sum_{j \in fanin(i)} \lambda_{j,i} = \sum_{h \in fanout(i)} \lambda_{i,h} \quad i = 1, ..., n$$

From this conservation of λ at each gate, the LRS objective is simplified to

$$\begin{aligned} Q(x) &= \sum_{i=0}^{n+1} \alpha_i x_i + \beta_i x_i^2 + \lambda_{0,0}(A_0 - R_0) + \lambda_{n+1,n+1}(-A_{n+1}) \\ &+ \sum_{i=1}^{n} \sum_{j \in fanin(i)} \lambda_{j,i} D_{ji} \end{aligned}$$

$Q(x)$ is a convex function so to find the optimal value of x_i, we set $\partial Q / \partial x_i = 0$. The optimal x_i is given as

$$\begin{aligned} x_i^{opt} &= -\Big\{\alpha_i + \sum_{j \in fanin(i)} \lambda_{j,i}(\hat{d}_{ji} + \hat{r}_{ji} \sum_{h \in fanout(i)} \tilde{c}_{ih}) \\ &+ \sum_{j \in fanin(i)} \sum_{k \in fanin(i)} \lambda_{k,j} \tilde{r}_{kj} \hat{c}_{ji} \Big\} / 2\beta_i \end{aligned}$$

Since $Q(x)$ is convex, considering the condition $L_i \leq x_i \leq U_i$, we get:

$$x_i = min(U_i, max(L_i, x_i^{opt}))$$

Lambda Update

To iteratively solve the dual problem, we use a multiplicative update:

$$\begin{aligned} \lambda_{j,i} = \lambda_{j,i} \times \frac{A_i}{A_j + D_{ji}} \qquad & i = 1, \ldots, n; j \in fanin(i) \\ \lambda_{0,0} = \lambda_{0,0} \times \frac{A_0}{R_0} & \\ \lambda_{n+1,n+1} = \lambda_{0,0} & \end{aligned}$$

We enforce λ conservation by normalizing $\lambda_{j,i}$ such that the ratio of all incoming $\lambda' s$ into i is preserved and the λ conservation condition is met.

IV.F.3 Computational Experience

The LR algorithm is implemented in 2500 lines of C++ code. We initialize λ by starting with a large initial value and then propagate it in reverse topological order. During propagation, λ at the output of a gate is divided equally among all the gate inputs to meet the lambda conservation condition. Static timing analysis is performed and coefficients of delays, input capacitance, and leakage are set according to the slew and loading capacitance of each cell. We use regression to derive the coefficients from standard-cell libraries.

We solve the LRS to find the optimum biases and then snap the biases to those available in the library. Static timing analysis is then run and delay, capacitance, and leakage model coefficients updated. Between iterations we update λ as

$$\lambda_{j,i} \quad = \quad \lambda_{j,i} \times \left\{ \frac{A_i}{A_j + D_{ji}} \right\}^a$$

where a reduces from 10 to 2, being multiplied by a factor of 0.98 in every iteration. This method of λ update results in faster convergence with no noticeable deterioration in the quality of results. Our multiplicative update is similar to [172].

Figures IV.11 and IV.12 show the cost of the primal problem and Lagrangian relaxation subproblem respectively as they change with iterations. As expected, the cost of the primal problem decreases while the cost of the subproblem increases; both costs saturate with the number of iterations. Table IV.20 compares the leakage reductions from Lagrangian relaxation and sensitivity-based downsizing for the following three testcases implemented with a $90nm$ library: c17 (13 instances), c432 (339 instances), c880 (703 instances). On a 2.0GHz AMD Opteron machine, our implementation took about 3.5 minutes on c880 and 16 minutes on c432. Runtime can be improved by techniques such as those of [172]. As seen from the table, Lagrangian relaxation yields appreciably better results than sensitivity-based downsizing. Additionally, Lagrangian relaxation allows a smooth tradeoff between delay and leakage that is not possible with a single run of our sensitivity-based downsizing approach.

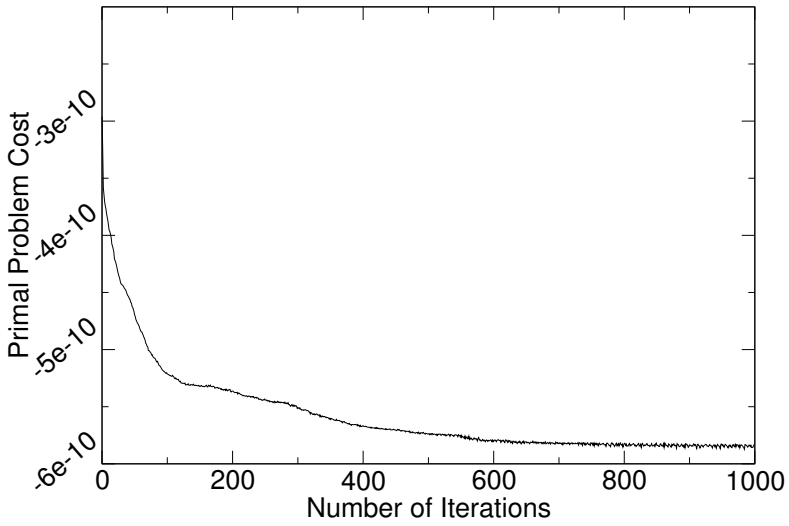


Figure IV.11: Cost of the primal problem with iterations.

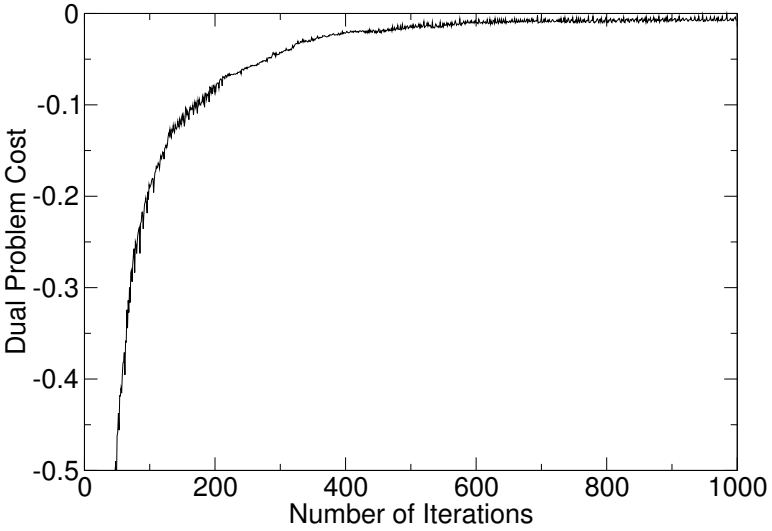


Figure IV.12: Cost of the Lagrangian relaxation subproblem.

Table IV.20: Leakage reductions for Lagrangian relaxation vs. sensitivity-based downsizing.

Circuit	Original Leakage (nW)	Sensitivity-Based		Lagrangian Relaxation	
		Leakage (nW)	Reduction (%)	Leakage (nW)	Reduction (%)
c17	148.5	133.0	10.44	131.2	11.65
c432	3947.8	3403.9	13.78	3363.5	14.80
c880	8709.9	7532.1	13.52	7442.3	14.55

IV.G On Synthetic Benchmarks with Known Upper Bounds

In the previous section we proposed a Lagrangian relaxation-based biasing algorithm to improve upon our greedy sensitivity-based optimization. Biasing falls in the general class of optimization problems known as gate sizing. In gate sizing, the sizes of cells in a circuit are determined to optimize a given objective under a set of constraints. Typical objectives are area, power, or hybrids thereof, and common constraints are path delay and slew constraints. A good solution to gate sizing minimizes circuit area and/or power while satisfying all requirements, and hence a high-quality solution is desirable.

Many approaches have been proposed for gate sizing under various objectives such as area and power (e.g., [62, 154, 54, 44, 153, 32, 165, 166, 89]). While some of these methods are understood to be suboptimal, others are expected to be optimal under the assumptions made. Unfortunately, many of these assumptions do not hold in practice and introduce suboptimality in methods that are claimed to be optimal. For example, gate delays are commonly assumed to be simple functions of gate sizes that have inadequate accuracy. Dependence of slew on cell size is often ignored, or delay is assumed to be a simple function of slew. Effects such as slew degradation over interconnects and resistive shielding of capacitance are gen-

erally not captured. As a result, all sizing methods are expected to be suboptimal in practice, and quantification of the suboptimality, to the best of our knowledge, has not be addressed.

Several studies have been conducted to understand the suboptimality of placement algorithms. Hagen et al. [78] stitched small designs to construct a larger design in such a way that the upper-bound on cost of the larger design was known. Chang et al. [38] developed suboptimality quantification that was outlined in [78] to construct synthetic benchmarks such that the optimal cost was known by construction. These benchmarks, however, were considered to be unrealistic and unrepresentative of real designs. To overcome this drawback, Cong et al. [56] added global hyperedges and established upper bounds. Kahng et al. [95] proposed a set of netlist transformations to a placed design that: (1) did not change the half-perimeter wire length (HPWL) of the placement, and (2) ensured that under any placement the HPWL of the original netlist was no more than of the transformed netlist. Thus, the HPWL of the initial netlist under the initial placement is a known upper bound for the placement of the transformed netlist.

We propose to extend the research developed for suboptimality quantification for placement and partitioning algorithms, to sizing algorithms. We believe that construction of benchmarks with known solution quality bounds is more challenging for the sizing problem than placement. While the objective for placement (wirelength) depends only on the placement for a given netlist and technology, the objective for sizing depends on the gate sizes as well as the available timing slack. In placement, once the locations of some cells are chosen, other cells are restricted to the remaining available locations. In sizing, however, the interdependence of sizes of different cells is more complex and depends on circuit topology, delay functions, and available slacks.

We now propose construction of testcases for evaluating sizing algorithms such that an upper bound on the minimum cost is known by construction. We construct simple, repetitive, and symmetrical circuit topologies for which high-quality results can sometimes be predicted by visual inspection, and refer to these testcases as “eye charts”. In addition to the circuit topologies, we also generate

Figure IV.13: Chain eye chart.

cell libraries with conveniently-chosen delay, area, and power values. Our eye charts, in addition to quantifying suboptimality, can provide valuable information that can be used to tweak and improve an optimization algorithm. We note that combinational eye charts are sufficient for suboptimality studies. This is because sequential circuits, for the purpose of sizing, can be transformed to combinational circuits by converting flip-flops to endpoints (with nodes for setup/hold delays and clock-to-data delays introduced), and are no more challenging than combinational circuits. I.e., an algorithm that performs well on combinational circuits can be trivially made to perform well on sequential ones.

IV.G.1 Chain Eye Chart

Our *chain* eye chart is shown in Figure IV.13. All cells are of the same type, A, and their output slew and input capacitance are assumed to be zero in the library. Furthermore, the wire delay is assumed to be zero, and cell delay is assumed to be independent of loading capacitance and input slew. Different sizes of the cell differ in terms of their delay, area, and/or power values. In the context of biasing, the library contains multiple variants of the cell A corresponding to different gate length biases. As the gate length increases, cell variants have linearly increasing delays and quadratically decreasing leakage values.

Several testcases can be generated corresponding to this eye chart by increasing the circuit slack. The slack can be increased in steps such that only a known number of cells can be sized. The optimal sizing solution can sometimes be identified by visual inspection, and can always be derived by mathematical programming. For example, in the biasing context in which delay is linear and leakage is quadratic with gate length bias, the following equations can be used to find the optimal biases.

$$\text{Minimize: } \sum_i l_1.x_i + l_2.x_i^2$$
$$\text{Such that: } \sum_i d_1.x_i \leq slack$$

where i is the index over topological level of the cells; x_i is the optimal bias assigned to the cell at topological level i; l_1 and l_2 are coefficients of leakage decrease with biasing; d_1 is the coefficient of delay increase with biasing; and *slack* is the circuit slack. The equations can be solved using integer programming or exhaustive enumeration if the number of cells in the testcase is not large. The difficulty of this eye chart can be increased by using different types of cells which differ in their delay and leakage coefficients (i.e., d_1, l_1, and l_2). The sensitivity-based biasing algorithm presented in Section IV.B performs optimally on this eye chart.

IV.G.2 Star Eye Chart

Figure IV.14 illustrates our *star* eye chart. In a manner similar to our chain eye chart, several testcases can be generated for this topology corresponding to different slack values. At least two types of cells must be used for this eye chart to support the different number of inputs. The two types of cells must, however, exhibit identical delay vs. objective (area or power) tradeoffs. Assumptions regarding slews, capacitance, and wire delays are identical to the chain eye chart.

Under the stated assumptions, the key observation is that all cells at a topological level must be sized identically in an optimal sizing solution. Visual inspection or mathematical programming can be used to identify the optimal sizing solution. For Figure IV.14 and in the context of gate biasing, the optimal biasing can be found from the following equations.

$$\text{Minimize: } \sum_i 3^{|i-2|}(l_1.x_i + l_2.x_i^2)$$
$$\text{Such that: } \sum_i d_1.x_i \leq slack$$

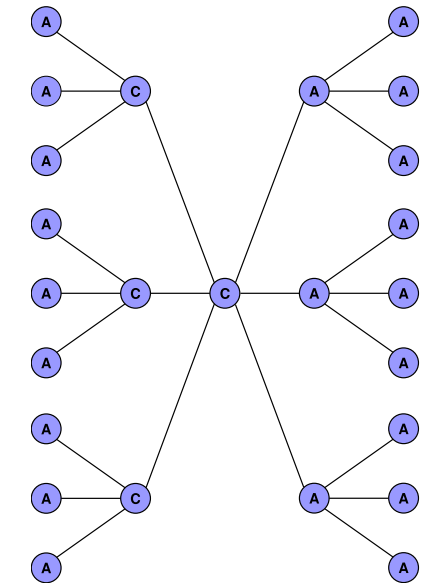


Figure IV.14: Star eye chart.

where all variables follow the nomenclature developed for the chain eye chart. This eye chart can be made more difficult by using multiple types of cells such that cells at different topological levels have different leakage and delay coefficients (cells at the same topological level must have the same coefficients). Our biasing algorithm presented in Section IV.B demonstrates substantial suboptimality on star eye charts. For the eye chart of Figure IV.14, when only one bias value is available, our algorithm biases one, three, or nine cells when only one level can be biased, while the optimal solution biases nine cells; our algorithm biases four, six, 12, or 18 cells when two levels can be biased, while the optimal solution biases 18 cells.

IV.G.3 Mesh Eye Chart

Our *mesh* eye chart is shown in Figure IV.15. This eye chart makes identical assumptions to our star eye chart. The figure shows the mesh eye chart in two dimensions but the topology can be extended to multiple dimensions. The key

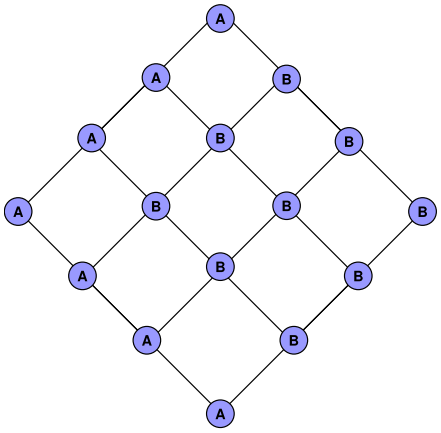


Figure IV.15: Mesh eye chart.

observation is that – similarly to our star eye chart – all cells at a given topological level must be sized identically in an optimal sizing solution. The optimal solution can be found with visual inspection or with mathematical programming, similar to the star eye chart.

For the eye chart of Figure IV.14, when only one bias value is permitted, our sensitivity-based algorithm biases one to four cells when only one level can be biased, while the optimal solution biases four cells; our algorithm biases two to seven cells when two levels can be biased, while the optimal solution biases seven cells. This eye chart can also be made more difficult by using cells that have different delay and leakage (or another objective) coefficients at different topological levels.

IV.G.4 Hybrid Testcases

The three eye charts can be combined to construct hybrid testcases. In particular, different eye charts of different sizes can be connected in parallel and in series. When connecting eye charts in series, the slack is budgeted arbitrarily among the eye charts. If several eye charts are connected in parallel, they must all have identical slack. Figure IV.16 illustrates a testcase in which our three eye charts are used and circuit total slack of $1ns$ is budgeted among them.

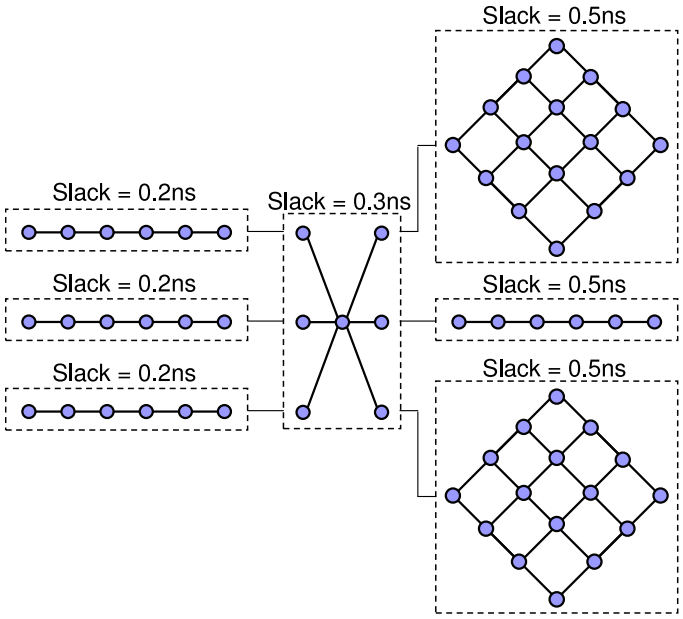


Figure IV.16: Hybrid testcase.

IV.H Conclusions

We have presented a novel methodology that uses selective, *small* gate length biases to achieve an *easily manufacturable* approach to runtime leakage reduction. Biasing can be implemented after signoff during RET. The changes made to layouts of the cells to derive their biased variants are transparent during the design flow. This transparency implies that the biased and unbiased cell layouts are physically identical and completely pin-compatible, and hence layout-swappable. Layout-swapability allows biasing-based leakage optimization to be possible at any point in the design flow, unlike sizing-based methods. For our testcases we observe the following.

- With a biasing of $8nm$ in a $130nm$ process, leakage reductions of 24% to 38% are achieved for the most commonly used cells with a delay penalty of under 10%.
- Using simple sizing techniques, we can achieve up to 33% leakage savings

with less than 3% dynamic power overhead and no delay penalty. Using more than two gate lengths for the most commonly used cells along with improved sizing techniques is likely to yield better leakage savings.

- We compare gate length biasing at the cell-level and at the transistor-level. Transistor-level gate length biasing can further reduce leakage by up to 10% but requires a significantly larger library. Therefore, transistor-level biasing should be done for only the most frequently used cells such as inverters, buffers, NAND and NOR gates. Fortunately, the most frequently used cells have one or two inputs and hence only a small number of transistor-level biasing variants needs to be characterized for them. For cells with three or more inputs, no transistor-level biasing variants may be created (i.e., only cell-level biasing variants are created). To further reduce library size, only one of the cell variants in which different logically equivalent inputs are fast may be retained, and pin-swapping techniques can be used during leakage optimization.

- The devices with biased gate length are *more* manufacturable and have a larger process margin than the nominal devices. Biasing does not require any extra process steps, unlike multiple-threshold based leakage optimization methods.

- Gate length biasing leads to more process-insensitive designs with respect to leakage current. Biased designs have up to 41% less leakage worst-case variability in the presence of inter-die variations as compared to nominal gate length designs. In the presence of both inter- and intra-die CD variations, selective gate length biasing can yield designs less sensitive to variations.

We have also studied simultaneous use of biasing and V_{th} assignment, and found the two techniques to complement each other. Biasing when used with dual-V_{th} can yield leakage savings similar to a costlier, triple V_{th} process. Foundry-set V_{th}'s may not yield the best leakage reduction, and the slack distribution, netlist structure, and leakage-delay tradeoff must be understood prior to V_{th} selection.

We also observe that the availability of biasing as an optimization knob, does not considerably change the set of threshold voltages that maximize leakage reduction subject to a delay constraint.

To improve the optimization quality, we adapt the Lagrangian relaxation-based sizing approach proposed in [44] to use for biasing. In comparison to our sensitivity-based optimization, which is a greedy approach, Lagrangian relaxation produces substantially better results in terms of leakage, for small examples. We highlight the need for quantification of suboptimality of gate sizing algorithms and present families of synthetic benchmarks ("eye charts") for which bounds on the optimal solution quality are easy to determine. Such benchmarks can be used to quantify suboptimality in sizing algorithms, and to identify algorithmic weaknesses for subsequent improvement.

IV.I Acknowledgments

This chapter is in part a reprint of:

- P. Gupta, A. B. Kahng, P. Sharma and D. Sylvester, "Gate-Length Biasing for Runtime Leakage Control," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25(8), 2006, pp. 1475 – 1485.

- P. Gupta, A. B. Kahng, P. Sharma and D. Sylvester, "Selective Gate-Length Biasing for Cost-Effective Runtime Leakage Control," *Proc. Design Automation Conference*, 2004, pp. 327 – 330.

- A. B. Kahng, S. Muddu and P. Sharma, "Impact of Gate-Length Biasing on Threshold-Voltage Selection," *Proc. International Symposium on Quality Electronic Design*, 2006, pp. 747 – 754.

I would like to thank my coauthors Prof. Puneet Gupta, Swamy Muddu, Prof. Andrew B. Kahng, and Prof. Dennis Sylvester.

V

Reducing CMP Variability Through Fill Insertion

V.A Introduction

CMP (Chemical-Mechanical Polishing) is the enabling technology to attain high levels of planarization [21]. As number of layers increases and linewidths shrink, tolerance for topographical imperfections decreases. This is due to tight depth-of-focus control requirements and high sensitivity of resistance to metal thickness. Despite improvements in CMP technology, layout pattern sensitivities are significant causing certain regions to have higher topographies than others due to differences in underlying densities [169]. Designers and manufacturers use techniques such as dummy fill insertion and slotting to respectively increase and decrease the metal density [96]. Dummy fills are non-functional features that do not directly contribute to the logic implementation. This chapter focuses on FEOL and BEOL fill. FEOL fill, also known as STI fill, is inserted to aid CMP that is performed after the oxide deposition step in STI to remove excess oxide. BEOL fill, also known as metal fill, is inserted into interconnect layers to aid CMP that removes excess oxide, copper, and barrier materials.

We first present a fill insertion methodology for FEOL fill and show that it improves planarity considerably. Traditional fill insertion is rule-based and is used

with reverse etchback [1] to attain desired planarization quality. Due to extra costs associated with reverse etchback, "single-step" STI CMP, in which fill insertion suffices, is desirable. Due to superior planarity with our fill insertion, the need for a reverse etchback process can be potentially eliminated. To alleviate the failures caused by imperfect CMP, we focus on two objectives for fill insertion: oxide density variation minimization and nitride density maximization. A linear programming based optimization is used to calculate oxide densities that minimize oxide density variation. Next a fill insertion methodology is presented that attains the calculated oxide density while maximizing the nitride density. In the results we show that our approach effectively attains low oxide density variation with high nitride density. Through CMP simulation, we show that our approach results in superior topography metrics.

In Section V.C, we present our studies of capacitive effects of BEOL fill and present certain guidelines that reduce these effects. It is well known that fill insertion adversely affects total and coupling capacitance of interconnects. While grounded fill can be extracted by full-chip extractors, floating fill can be reliably extracted by small-scale 3D field solver simulations only. Due to a poor understanding of the impact of floating fill on capacitance, designers insert floating fill conservatively. We study the impact of floating fill insertion on coupling and total capacitance when the fill geometry and both the interconnects between which the capacitance is measured are on the same layer. We show that the capacitance with the same-layer neighboring interconnects is a large fraction of total capacitance, and that it is significantly affected by fill geometries on the same layer. We analyze the effect of fill configuration parameters such as fill size, fill location, interconnect width, interconnect spacing, etc. and consider edge effects and effects occurring due to insertion of several fill geometries in close proximity. Based on our findings, we propose certain guidelines to achieve high metal density while having a smaller impact on interconnect capacitance. Finally, we validate the proposed guidelines using representative process parameters and a 3D field solver.

[1] Reverse etchback [109] uses an extra mask to etch away oxide from high oxide density regions and make the oxide density more uniform.

V.B FEOL Fill for Improved Planarity

STI is the mainstream CMOS isolation technique used in all designs today. In STI, trenches are created in the silicon substrate and filled with silicon dioxide (oxide) around devices or groups of devices that need to be isolated. Advanced STI processes involve many process steps of which nitride deposition, oxide deposition, and CMP are of interest. Nitride is deposited over active regions to protect the underlying silicon and to act as a polish stop. In areas outside the active regions, trenches are created and void-free oxide is deposited over the wafer by chemical vapor deposition (CVD). CMP is used to remove the excess oxide over the nitride and trenches to ensure a planar surface for successive process steps.

CMP is the planarization technique of choice and is used extensively in IC fabrication processes for metal layers and for STI. In CMP for STI, deposited oxide is removed until all oxide over the nitride regions is removed. Unfortunately, due to high pattern dependency, CMP is imperfect and, depending on the underlying patterns, can result in functional and parametric yield loss. The pattern densities of both the deposited oxide and the underlying nitride determine the planarization quality after CMP. Because oxide is deposited over nitride, oxide density is dependent on the shapes of the underlying nitride features as explained in the next section. Therefore the density and the shapes of the nitride features determine the planarization quality. Traditionally, planarity imperfections have been addressed by reverse etchback and by fill insertion. In reverse etchback, a second mask is created to etch away oxide in regions of high oxide density prior to CMP, resulting in a more uniform oxide density. Unfortunately, an extra mask and additional process steps are required for reverse etchback and it is economically desirable to avoid reverse etchback. Fill insertion is another technique to control oxide and nitride densities. Fill insertion for STI CMP involves adding dummy nitride features to increase the nitride and, through it, the oxide density.

Typically, rule-based fill insertion is performed by shape-based tools such as Mentor Calibre. Dummy rectangles are tiled with a predefined size, spacing, and keep-off distance from the design's features. Often this approach is used to

control only the nitride density along with reverse etchback which controls the oxide density. Beckage et al. proposed a model-based fill insertion methodology that uses CMP simulation, an area of active research [191, 23, 108], to identify regions for fill insertion [22]. Their approach uses two types of fill "tiles": (1) tiles that contribute to the nitride density but negligibly to the oxide density, and (2) tiles that contribute to both oxide and nitride densities. Post-CMP topography simulation is then used to drive the insertion of these tiles in the layout. Topography simulation is based on complex models and the determination of the oxide and nitride densities for the desired topography is also complicated. Unfortunately, details are not provided by the authors. Also, due to the use of specific fill configurations (tiles), the flexibility to control densities is limited. We propose a fill insertion methodology that targets oxide density variation minimization and nitride density maximization. These two objectives help alleviate the failures caused by CMP imperfections as discussed later.

We first apply a linear programming-based optimization that was previously proposed for BEOL CMP [97] to calculate target oxide densities that minimize the oxide density variation. With the target oxide densities determined, fill insertion is performed to maximize nitride density. We insert fill wherever permitted by the design rules and then remove it on-demand to meet the target oxide density. We develop an algorithm to attain the target oxide density by removing a minimum amount of fill (so that nitride density is maximized). We evaluate the proposed approach on two large testcases. Compared to the unfilled layout and layout with fill tiling, we observe that our proposed approach has a substantial reduction in oxide density variation as well as an enhancement in nitride density. Further, we run a CMP simulation to predict the post-CMP topography. We find that the topography achieved for the layout with the proposed methodology has superior characteristics. We also hypothesize that stress due to STI decreases when fill is inserted with the proposed approach.

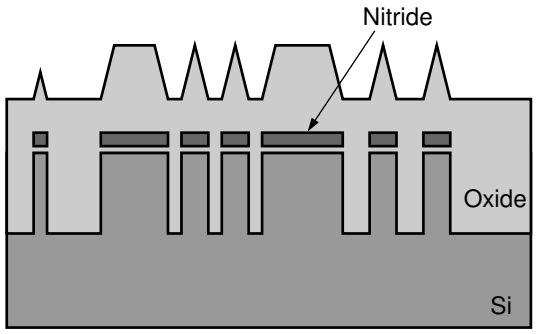


Figure V.1: Profile before CMP. Oxide is deposited with slanted sidewalls over nitride features.

V.B.1 Background

The basic STI process steps are as follows. First oxide, known as pad oxide, followed by nitride is deposited over the wafer. Then the deposited nitride is patterned and allowed to remain only over the active (or diffusion) regions. Everywhere else trenches are etched into the silicon and then oxide deposited by CVD over the wafer. Though the oxide is deposited to fill the trenches, it also deposits over the nitride features and is called *overburden* oxide. Figure V.1 shows a cross-section after these steps.

CMP is used to planarize the surface for successive process steps. Figure V.2 shows the desired cross-section after CMP. In reality, however, such a planar cross-section is not attained. Imperfect planarization can result in three key failure mechanisms shown in Figure V.3 [28]. First, if the oxide over *all* nitride regions is not completely cleared, then subsequent stripping of nitride will be prevented, leading to device failure. Second, excessive polishing causes nitride erosion which leads to a lowered isolation edge and consequently poor device characteristics. Excessive nitride erosion can also cause stripping of underlying silicon and device failure. Third, oxide in larger trenches dishes due to pad-bending, causing poor isolation.

The primary requirements of CMP are: (1) complete removal of oxide over all nitride regions and (2) no stripping of silicon under the nitride. These two requirements determine the *planarization window*, which is the time interval from

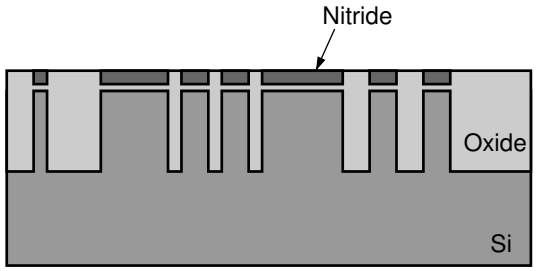


Figure V.2: Desired profile after CMP. Oxide over nitride should be completely cleared, no nitride should erode, and no oxide dishing should occur in the trenches.

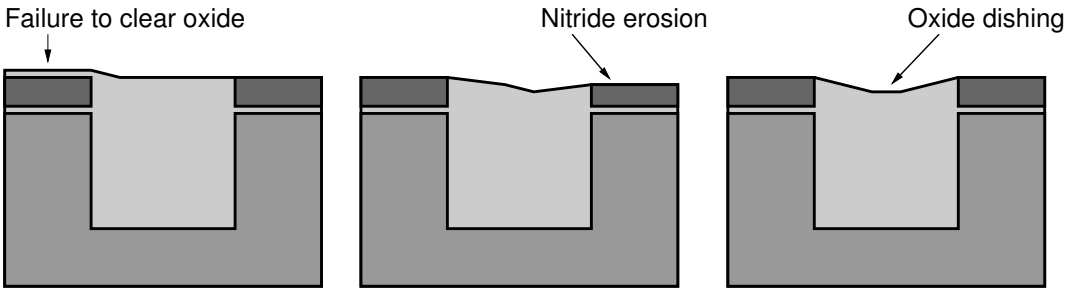


Figure V.3: Three key failure mechanisms caused by imperfect CMP.

the instant when all oxide over nitride just gets removed to the instant when silicon at any location is touched by the pad. Planarization can only be stopped at a time instant in the planarization window and it is desirable to have a large planarization to accommodate for variations. In addition, oxide dishing and nitride erosion must be minimized for improved device characteristics.

In STI CMP, post-planarization topography is affected by the density of the overburden oxide that is polished and that of the underlying nitride. Interestingly, because oxide is deposited over nitride, oxide density is dependent on the underlying nitride features. For high density plasma (HDP) oxide deposition, which is the mainstream oxide deposition technology, the deposition profile exhibits a slanted sidewall. Consequently, features on the oxide layer appear as *shrunk* nitride features [141, 23, 191]. Specifically, a nitride polygon is shrunk or sized down by a fixed amount (denoted by α) on each side to get the oxide polygon deposited over it. For example, nitride squares that are 5α on a side appear on the oxide layer as squares of side 3α, while squares of side less than 2α do not appear on the oxide layer. We note that shrinkage by α on all sides is a convenient approximation and accounts for sidewall slant and pad bending effects. Shrinkage allows us to control oxide and nitride densities independently up to some extent.

Fill insertion is performed by inserting features on the nitride layer to control densities of oxide and nitride layers. Design rules such as minimum nitride width and area, maximum nitride width, minimum nitride spacing and notch, and minimum enclosed area by nitride must be followed in fill insertion. Inserted fill is always separated by the minimum nitride-to-nitride spacing from all design features. So even after fill insertion there is a trench to isolate the design features ensuring negligible electrical impact of the inserted fill. Since there are no contacts with the inserted fill, no stray devices that can potentially act as parasitics are formed. Moreover, no diffusion may be done over the fill features. Fill insertion can potentially affect stress induced by STI as explained in Chapter III. Stress affects device characteristics because of its impact on carrier mobility and is modeled, in part, in today's device models (e.g., BSIM v4.4.0) [42]. Recently, STI fill insertion was noted to improve predictability of stress-induced effects and therefore reduce

guardbanding [127]. We propose a methodology that inserts fill and performs placement perturbations to alter stress and improve circuit timing in [100].

V.B.2 Motivations and Objectives of Fill Insertion

In this section we present the motivation behind fill insertion for STI and formulate the objectives of fill insertion. Fill insertion is used to attain a more uniform density, and to consequently reduce the topography variations after CMP which is pattern dependent. The primary goal of fill insertion is to maximally reduce causes for three key manufacturing failures due to imperfect CMP – failure to clear oxide on top of nitride, nitride erosion, and oxide dishing (see Figure V.3). The secondary goal of fill insertion is to control STI-induced stress, a significant component of which is not modeled due to the size of STI wells. With fill insertion, the size of STI wells around devices can be made consistent to increase the accuracy of device performance and power estimates.

Failure to clear oxide is the primary cause of CMP failure. It occurs in regions where oxide density is substantially higher than average. Therefore oxide density variation must be minimized. Reduction of oxide density variation is also beneficial for reduction of another type of CMP failure. Since more oxide over nitride can be cleared simultaneously, the size of the planarization window can be increased which results in reduction of nitride erosion.

Oxide dishing and nitride erosion can be greatly reduced by increasing nitride density. Indeed, higher nitride density results in smaller trenches and, therefore reduces oxide dishing. The mechanism of reduction of nitride erosion is based on the fact that nitride is significantly harder than oxide. When the polishing pad touches the nitride surface, increased load on the driving motor is detected and polishing stops. Obviously, higher nitride density makes the detection of the nitride level more accurate.

As described in Chapter III, STI stress is due to: (1) size of diffusion regions and (2) size of the STI well isolating the diffusion regions. Stress due to diffusion size is already included in today's SPICE models. However, stress due to

STI well size is not yet modeled and can be a significant source of variation [127]. Typical power/performance characterization considers wells of smallest or largest size for the best- and worst-case estimates. When nitride density is higher, then devices get smaller STI wells around them which reduces the difference between these estimates, which in turn makes their power/performance more predictable.

The above analysis leads to the following two objectives for fill insertion in order of their priority:

1. Minimize oxide density variation.
2. Maximize nitride density.

The corresponding bi-criteria problem formulation is described in the next section. In Section V.B.4, this problem is transformed into the problem of nitride density maximization subject to an upper bound on the oxide target density.

V.B.3 Bi-criteria Formulation and Optimization for Fill Insertion

Given:

- set of rectilinear nitride regions contributed by the devices in the design;
- parameter α by which nitride features shrink on each side to give oxide features; and
- design rules: minimum nitride width, maximum nitride width, minimum nitride space and notch, minimum nitride area, minimum enclosed area by nitride.

Find:

- locations for fill insertion.

Such that:

1. oxide density variation is minimized; and
2. nitride density is maximized.

The above bi-criteria formulation has clear prevalence of the first objective over the second. Therefore, we first address the primary objective: oxide density variation minimization and afterwards maximize nitride density in such a way that the first objective is not affected.

Formally, *density variation* is defined as the maximum difference in densities computed over fixed-sized windows of the layout [97]. Figure V.4 shows overlapping windows over which density is computed. Tile size is the distance by which the windows are offset from each other. The *fill synthesis* problem for minimum density variation can be formulated as follows:

Given:

- *fill slack*, s_i, the maximum amount of fill that can be inserted in Tile i, without any DRC violations; and
- *window size*, r, as a multiple of tile size, over which density is computed.

Find:

- *target fill*, t_i, the amount of fill to be inserted in Tile i.

Such that:

- density variation is minimized.

The fill slack for the STI technique is equal to the maximum oxide density contributed by fill insertion. We observe that the maximum contribution is made by maximum fill insertion on the nitride layer (i.e., insert fill wherever possible). The *maximum fill region*, the union of all regions where fill can be inserted subject to DRC constraints, is denoted by $Nitride_{max}$ (density = $|Nitride_{max}|$).

The procedure for finding the region $Nitride_{max}$ is illustrated on Figure V.5. The nitride regions contributed by the devices in the design are shown in Figure V.5(a). First, to obey the minimum spacing design rule, the features are bloated by the minimum spacing. Minimum spacing design rule-correct fill may be inserted in the remaining regions (Figure V.5(b)). Next, to obey the minimum nitride width and area rules, regions that are too small are removed (Figure V.5(c)). $Nitride_{max}$ is the region available for fill insertion after these two steps.

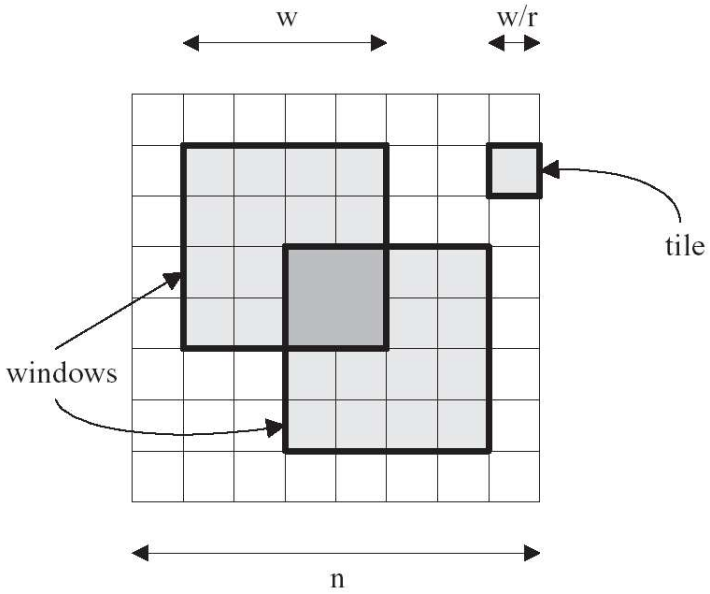


Figure V.4: Layout is partitioned into windows of fixed size $w \times w$ and density is computed over them. Density variation is the maximum difference between densities computed over any two windows.

Maximum oxide density contribution is found by shrinking $Nitride_{max}$ by α on all sides. We use $|Oxide_{max}|$ to denote the oxide density from $Nitride_{max}$, which is the highest oxide density achievable by fill insertion.

We use the linear programming based solution proposed in [97] to synthesis fill for the minimum density variation problem. Other approaches such as Monte-Carlo method-based, greedy, and hybrid approaches can also be used [48]. These solutions find the target oxide density per tile.

V.B.4 Nitride Maximization Formulation and Optimization

The bi-criteria problem statement can be transformed into the following:

Given:

- set of rectilinear nitride regions contributed by the devices in the design;

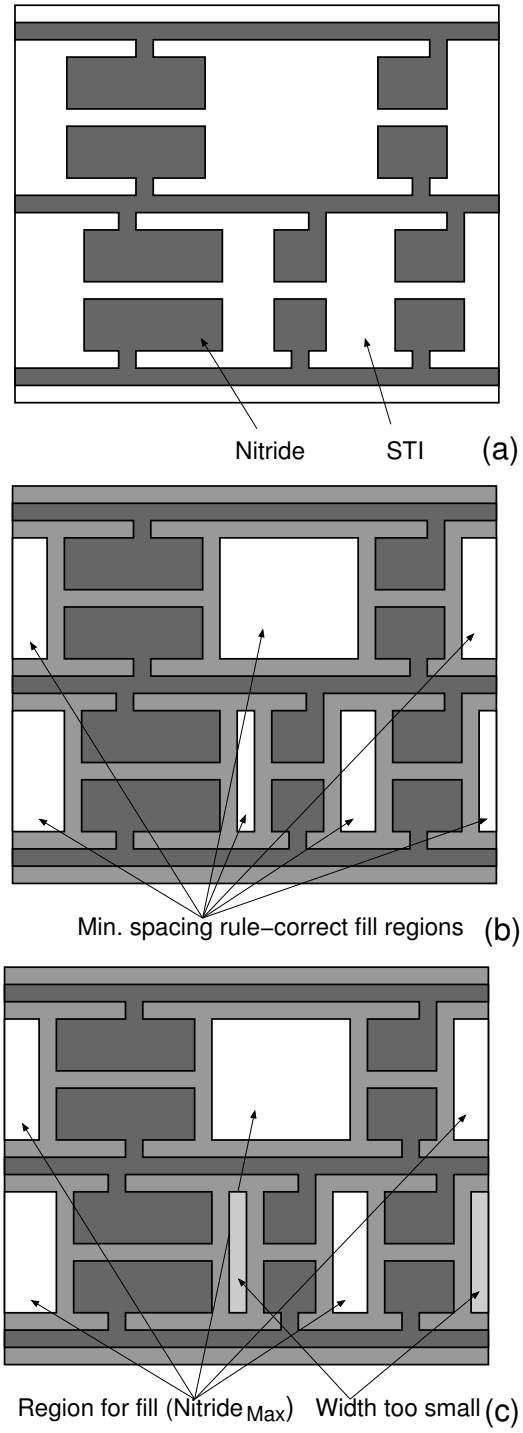


Figure V.5: Computation of maximum fill region ($Nitride_{max}$). (a) Unfilled layout. (b) Design features bloated by minimum spacing design rule. (c) Spaces of small width and area (illustrated in the lightest shade of gray) are not available for fill.

- parameter α by which nitride features are shrunk on each side to give oxide features;
- design rules: minimum nitride width, maximum nitride width, minimum nitride space and notch, minimum nitride area, minimum enclosed area by nitride; and
- *target oxide density per tile.*

Find:

- locations for fill insertion

Such that:

- nitride density is maximized.

Proposed Solution

We first consider the following two important limiting cases of $|Oxide_{target}|$:

1. $|Oxide_{target}| = |Oxide_{max}|$
2. $|Oxide_{target}| = 0$

Case $|Oxide_{target}| = |Oxide_{max}|$. This is the trivial case. Fill is inserted at $Nitride_{max}$ to attain oxide density of $|Oxide_{max}|$ and nitride density of $|Nitride_{max}|$. We note that the maximum nitride size design rule is typically over $100\mu m$ which is significantly larger than typical lengths of polygons in $Nitride_{max}$. Therefore, we ignore the maximum nitride size design rule for computing $Nitride_{max}$; any DRC violations are fixed post-fill.

Case $|Oxide_{target}| = 0$. We note that due to the nature of the problem, there is no need to increase the oxide density of many tiles and this case is very frequent. For this case, nitride fill features that do not contribute to the oxide density must be inserted. Fill rectangles that have one side smaller than 2α do not contribute to the oxide density due to shrinkage by α on each side. Unfortunately, rectangular fill features are suboptimal in offering the highest nitride density. To have zero oxide density, all points on inserted fill shapes must be within a distance α from the nearest edge of the shape. We first insert fill at $Nitride_{max}$ and then *dig* holes of minimum size in the fill to ensure all points on fill are within a distance α from the nearest edge, i.e., no density is contributed to oxide.

Lemma 1 *Fill at $Nitride_{max}$ with rectangular holes of minimum combined area, such that: (1) all points on fill are within a distance α from an edge and (2) hole size is no smaller than that permissible by DRCs, offers the highest nitride density with zero oxide density.*

Proof. Due to shrinkage by α on each side, no point on the nitride contributes to the oxide density. The oxide contribution is therefore zero. All rectilinear nitride fill configurations can be realized with fill at $Nitride_{max}$ with rectangular holes. Minimization of hole area is equivalent to nitride density maximization. □

We refer to the area on nitride that is within a distance α of a hole as the *area covered* by the hole. Area covered by a hole does not contribute to the oxide density.

Lemma 2 *Highest area is covered per unit hole area by holes that are square in shape and of the smallest size permissible by DRCs.*

Proof. Figure V.6 shows a hole and the area covered by it. The area covered by a hole of size $a \times b$ is $\pi\alpha^2 + 2a\alpha + 2b\alpha$. The ratio of area covered and the hole size is $(\pi\alpha^2 + 2a\alpha + 2b\alpha)/(ab)$ and is the highest for the square hole of the smallest size. □

Lemmas 1 and 2 suggest the following strategy: (1) insert maximum fill in the entire region $Nitride_{max}$ and (2) dig the minimum number of smallest-sized squared holes in this region. The smallest size of squared holes is determined by the minimum diffusion-diffusion spacing rule and/or the minimum diffusion notch rule. We denote the minimum hole size by β. For zero oxide contribution we must ensure that the entire $Nitride_{max}$ region is covered with the rounded squares. In addition, the overlap between rounded squares should be minimized to require the minimum number of holes. The problem is essentially the known *covering* problem in computational geometry.

Unfortunately rounded squares are difficult to handle in covering and must be simplified to a shape that is more amenable to the covering problem. Triangles, rectangles and hexagons are such shapes. Several other polygons such as

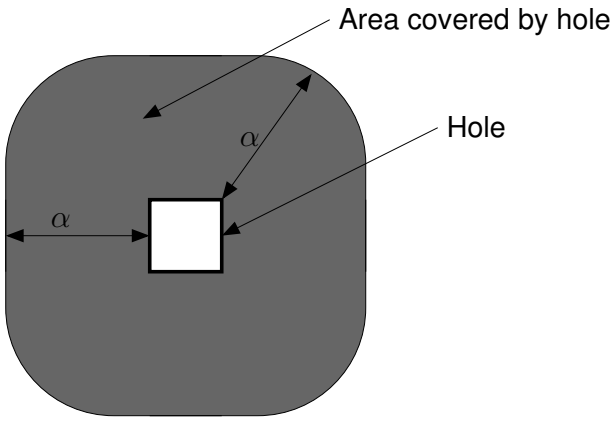


Figure V.6: Gray area is the *area covered* by the white hole, i.e., fill features added in the gray area do not contribute to the oxide density due to the hole. α is the shrinkage; oxide features can be computed from nitride features by shrinking by α on all sides.

pentagons, heptagons, and octagons require substantial overlap for covering. The simplified polygon must be completely inscribable within the rounded square and then covering done with the simplified polygon. Due to this simplification, not all area offered by the rounded square will be used for covering. The area of the rounded square that is outside the inscribed simplified polygon is referred to as the *inloss*. Figure V.7 shows an inscribed hexagon and the associated inloss. We wish to use the polygon that offers the minimum inloss. Triangles, clearly, have a larger inloss in comparison to rectangles and hexagons. We use hexagons, that are similar to regular hexagons but allow two parallel edges to be of different lengths than the other four, for covering. We refer to such hexagons as *parallelohexagons* because opposite edges are parallel. Parallelohexagons are more flexible than regular hexagons and better for covering. Parallelohexagons are flexible enough to be reduced into rectangles so covering with parallelohexagons is no worse than with rectangles.

We now calculate the best parallelohexagon given a rounded square of parameters α and β. As the rounded square is symmetrical about the X- and Y-axes, only the orientation in Figure V.7 and those generated by it after up to 45^o of rotation need to be evaluated. It may be shown that the smallest inloss is attained

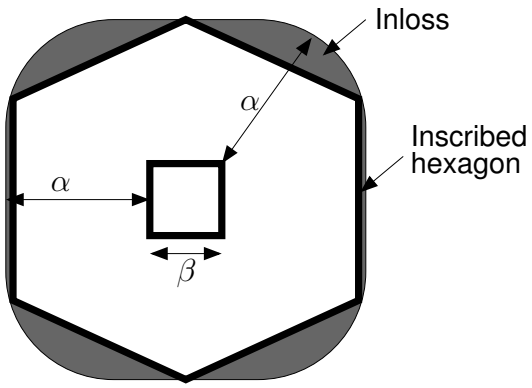


Figure V.7: Hexagon inscribed in a rounded square and the associated inloss (shown in gray). β is the minimum hole size permitted by the design rules.

in the orientation of Figure V.7 and when one vertex of the parallelohexagon is on the top edge of the rounded square and another on the bottom. The area of the parallelohexagon whose X-coordinate of the two rightmost vertices is x, is denoted by $A(x)$.

$$A(x) = \frac{1}{2}\Big[x\sqrt{\alpha^2 - (x-\beta/2)^2} + \alpha x + \beta x\Big]$$

$$\begin{aligned}\frac{dA(x)}{dx} &= \frac{1}{2}\Big[\sqrt{\alpha^2 - (x-\beta/2)^2} + \frac{x(x-\beta/2)}{\sqrt{\alpha^2 - (x-\beta/2)^2}} + \beta + \alpha\Big] \\ &= \frac{1}{2\sqrt{\alpha^2 - (x-\beta/2)^2}}\Big[\alpha^2 + \frac{1}{2}\beta x - \frac{1}{4}\beta^2\Big] + \frac{1}{2}(\alpha+\beta)\end{aligned}$$

From the derivative it is clear that the parallelohexagon area increases with x. Therefore the parallelohexagon with the minimum inloss has all its vertices on the rounded square. The corresponding inloss is given by $\{\alpha\beta+(\pi-2)\alpha\}/\{\beta^2+4\alpha\beta+\pi\alpha^2\}$ and is under 10% for typical values of α and β.

Covering rectilinear regions with parallelohexagons

We now present our algorithm to cover $Nitride_{max}$ which is rectilinear in shape with parallelohexagons that represent the area covered by holes. We overlay

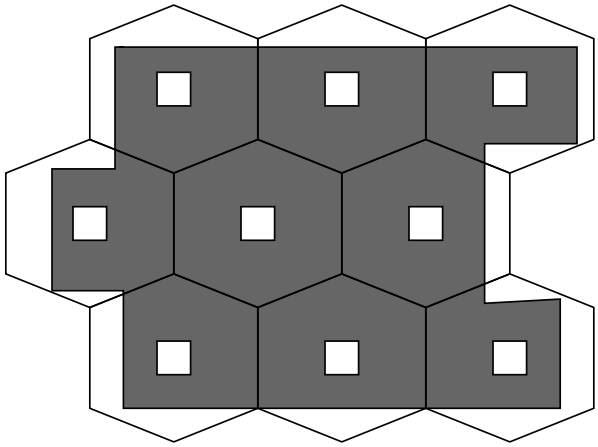


Figure V.8: Gray rectilinear polygon represents $Nitride_{max}$. Transparent hexagons are tessellated in a honeycomb to cover the polygon with a minimum number of hexagons. Holes created in $Nitride_{max}$ at the center of the hexagons (shown in white) ensure zero oxide density contribution due to $Nitride_{max}$.

a honeycomb structure which is a tessellation of parallelohexagons on the rectilinear polygon such that a minimum number of hexagons are required in the honeycomb. A honeycomb overlay that completely covers the rectilinear polygon and requires the minimum number of hexagons is referred to as an *optimal* overlay. To propose an algorithm for optimal overlay, we develop the following terminology. As shown in Figure V.9(a), we define V-segments, LH-segments, and UH-segments of a rectilinear polygon as its vertical edges, horizontal edges which have the polygon over them, and horizontal edges which have the polygon under them. Figure V.9(b) shows V-segments, LH-segments, and UH-segments of a honeycomb structure.

Theorem V.1 *In an optimal overlay:*

- *at least one V-segment of the honeycomb must align horizontally with a corresponding segment from the rectilinear polygon; and*
- *at least one LH- or UH-segment of the honeycomb must align vertically with a corresponding segment from the rectilinear polygon.*

Proof. Given an optimal overlay, the honeycomb can be perturbed to horizontally align *one* V-segment of the honeycomb with that of the rectilinear polygon, and

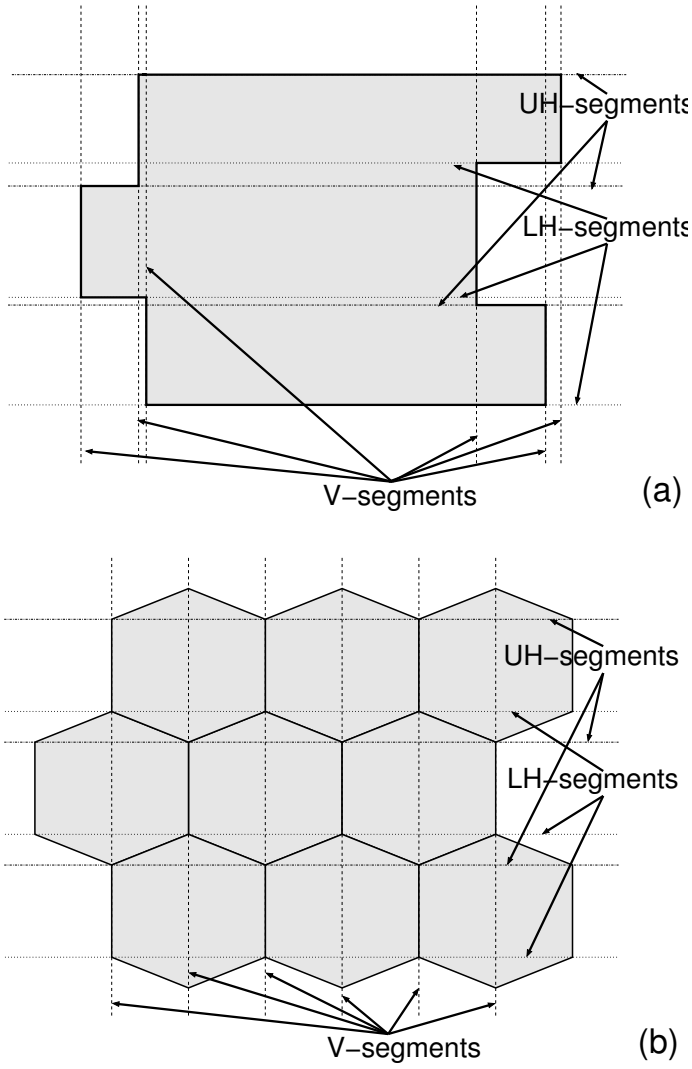


Figure V.9: Illustration of V- (vertical), LH- (lower horizontal), and UH- (upper horizontal) segments for a (a) rectilinear polygon, and (b) honeycomb.

vertically align *one* of LH- or UH-segment of the honeycomb with that of the rectilinear polygon, without requiring any additional hexagons to cover. Hence, there is an optimal overlay for which at least one V-segment of the honeycomb is horizontally aligned with a corresponding segment from the rectilinear polygon, and at least one LH- or UH-segment of the honeycomb is vertically aligned with a corresponding segment from the rectilinear polygon. Hence proved. □

Our algorithm to find the optimal overlay is as follows. Select one V-segment and one LH- (UH-) segment of the honeycomb, and one V-segment and one LH- (UH-) segment of the honeycomb. Horizontally align the V-segment and vertically align the LH- (UH-) segment to fix the position of the honeycomb over the rectilinear polygon. Count the number of hexagons required to cover the polygon. Iterate over all combinations of V- and LH- (UH-) segments to find the one with the minimum number of hexagons. To evaluate overlays in which the honeycomb is rotated by 90^o, the polygon is rotated by 90^o and the algorithm repeated. We do not consider other orientations of the honeycomb since only axes-aligned holes can be created. The complexity of the algorithm is $|PolygonV\_segments|$ $\times$ ($|PolygonLH\_segments| + |PolygonUH\_segments|$) $\times$ $|PolygonArea|$, where $|PolygonV\_segments|$, $|PolygonLH\_segments|$, $|PolygonUH\_segments|$, and $|PolygonArea|$ are the number of V-segments, number of LH-segments, number of UH-segments, and area of the polygon.

General Case $0 < |Oxide_{target}| < |Oxide_{max}|$. Due to the nature of the linear programming solution [97], tiles which require density increase get an $|Oxide_{target}| = |Oxide_{max}|$ and the general case is very infrequent. As in the previous case, we first perform fill insertion in $Nitride_{max}$ and then create holes of the minimum size since they offer high nitride density with zero or small oxide density. An area covered by holes, which is shaped as a rounded square, is approximated by parallelohexagons.

However, unlike the previous case, it is not necessary to cover the rectilinear polygon with hexagons. To ensure coverage in the previous case, rounded squares were approximated with *inscribed* parallelohexagons which caused the rounded square area outside the parallelohexagon to overlap and therefore required

more holes. Since covering the polygon is no longer necessary, we approximate rounded squares with *circumscribed* parallelohexagons. Packing the polygon with such parallelohexagons ensures no overlap between covers of two holes and requires fewer holes. Unlike the previous subsection, each parallelohexagon contributes to the oxide density in the regions that lie outside the rounded square but inside the parallelohexagon. We use the parallelohexagon of the smallest area so that its oxide density contribution is small; oxide density can easily be increased by not creating holes as described later. With an iterative program, we find that the smallest parallelohexagon is less than 8.9% larger than the rounded square (Figure V.10). We refer to the ratio of the contributed oxide area to the parallelohexagon area as *outloss*. I.e., $outloss = (area_{hexagon} - area_{roundedsquare})/area_{hexagon}$. Depending on the outloss, we now consider two cases:

1. $|Nitride_{max}| \times Outloss \leq |Oxide_{target}|$.
   I.e., if $Nitride_{max}$ was packed with the circumscribed hexagons, resultant oxide density would be less than $|Oxide_{target}|$. We use the parallelohexagon covering algorithm proposed earlier to overlay a honeycomb of circumscribed hexagons over a rectilinear polygon. Hexagons are then removed from the honeycomb, in decreasing order of their area outside the rectilinear polygon, until oxide density = $|Oxide_{target}|$.

2. $|Nitride_{max}| \times Outloss > |Oxide_{target}|$
   We partition the rectilinear polygon into two rectilinear polygons such that the area of the first, $A_1 = |Oxide_{target}|/Outloss$. In the first polygon, circumscribed hexagons are overlayed using the covering algorithm previously described. In the second polygon, which requires zero oxide density, we use solution of the $|Oxide_{target}| = 0$ case.

V.B.5 Experimental Study

We now describe our empirical validation of the proposed methodology. In the experiments we start with the design layout, and insert fill with the rule-

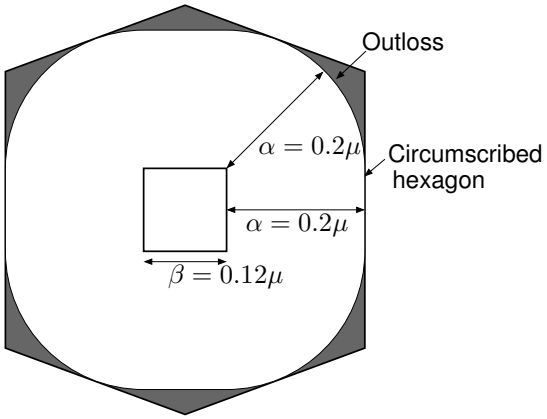


Figure V.10: Smallest hexagon circumscribed around the rounded square. The gray area represents the outloss.

based tiling and with the proposed approaches. Comparisons are then performed between: (1) the original layout, (2) layout after tiling-based fill insertion, and (3) layout after fill insertion performed with the proposed methodology. Our comparison studies are of two types: (1) analysis of oxide and nitride densities and (2) assessment of the post-CMP topography as predicted by a CMP simulator.

For the experiments, we create two large designs by assembling smaller cores. Commercial EDA tools with Artisan TSMC $90nm$ libraries and layouts are used for synthesis and placement of these circuits. Since interconnects do not affect nitride and trench regions, no routing was performed. We keep the utilization ratio between 60% and 70% which is typical. The first testcase, *mixed*, is composed of a RISC processor, a JPEG compressor, and AES and DES3 encryption cores. The design contains static memory and 756K cells, and measures $2mm \times 2mm$. The second design, *OpenRisc8*, is composed of eight RISC processor cores, contains static memory and 423K cells, and measures $2.8mm \times 3mm$.

Figure V.11 shows a small section of *OpenRisc8*. Figure V.11(a) is the unfilled layout with nitride in the shaded rectilinear regions and trenches everywhere else. The same section after tiling-based fill insertion (fill size = 0.5μ, fill spacing = 0.5μ) performed with *Mentor Calibre v9.3_5.9* is shown in Figure V.11(b). Fill regions are illustrated in gray. In Figure V.11(c) the same section with fill insertion

performed with the proposed methodology is shown. As is evident, nitride density is substantially higher with the proposed fill approach. Holes created in fill regions to control the oxide density are also visible.

Analysis of Nitride and Oxide Densities

The proposed methodology is driven by oxide and nitride density objectives that largely determine post-CMP planarity. The two objectives of our approach are oxide density variation minimization and nitride density maximization. Table V.1 presents the maximum oxide density variation, minimum nitride density, and average nitride density. In all our experiments, density is computed in overlapping square windows of side 160μ; the offset between successive windows is 40μ. For tiling-based fill insertion, we consider three fill-width/fill-spacing combinations: $0.5\mu/0.5\mu$, $1.0\mu/0.5\mu$, and $1.0\mu/1.0\mu$. It is clear that fill insertion with the proposed approach significantly decreases the oxide density variation and increase the nitride density. Compared to $0.5\mu/0.5\mu$ tiling-based fill, oxide density variation reduces by 63% and minimum nitride density increases by 79% when averaged over the two testcases. We also observe that tiling-based fill may increase the oxide density variation, which in turn requires costly etchback process steps to reduce it.

Post-CMP Topography Assessment

The density results show that the proposed approach achieves its objectives well. However, since the real goal of fill insertion is improved post-CMP planarity, it is important to assess that. We use the STI CMP simulator developed and calibrated by MIT's MTL group [108, 191] to predict post-CMP topography. Typical values are used for the initial structure and CMP model parameters, such as planarization length, pad bending, slurry selectivity, etc. We study the two primary characteristics of CMP quality - planarization window and final step height. Planarization window is the time window in which polishing may be stopped. If polishing is stopped earlier, oxide still remains over the nitride. If polishing is stopped later, the underlying silicon is stripped. Both these effects can lead to device failure. It is desirable to have a large planarization window to accommodate

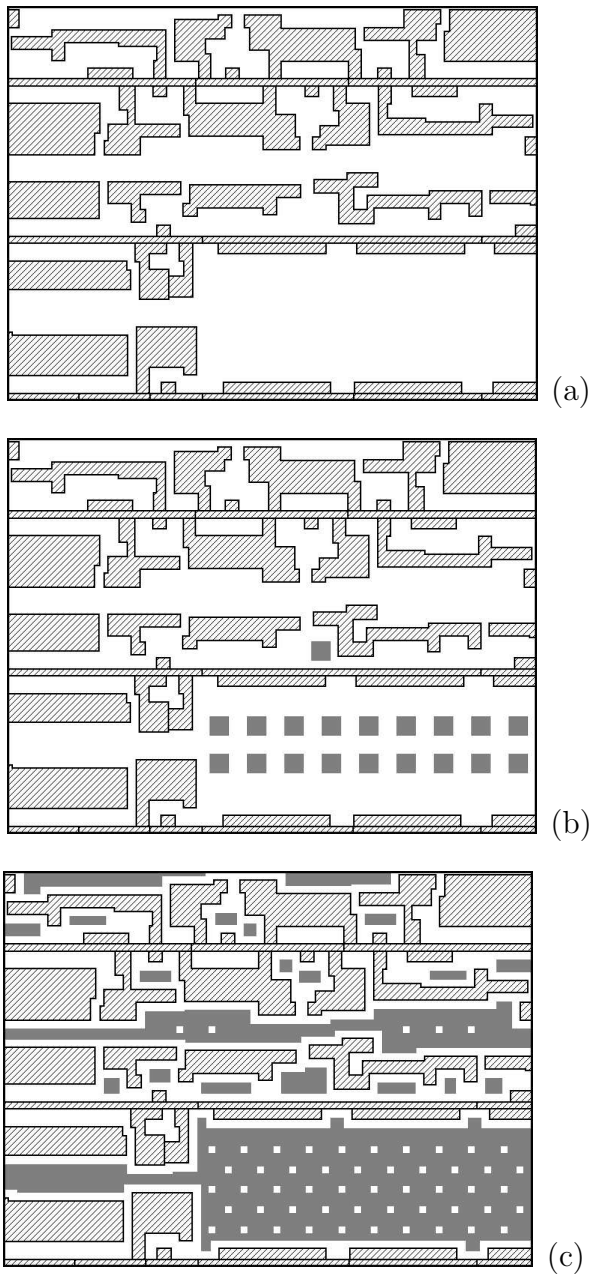


Figure V.11: Layout with fill inserted using tiling-based method and with the proposed method. Unfilled layout (a), layout with tile-based fill inserted (b), and layout with fill inserted with the proposed method (c) are shown. Fill is shown in gray and the shaded regions represent nitride due to CMOS devices (i.e., diffusion regions).

Table V.1: Density improvements from the proposed fill insertion method. Oxide density variation, minimum nitride density, and average nitride density are compared for two testcases for the unfilled layout, layout with tiling-based fill for three fill-width and fill-spacing combinations, and layout with fill inserted using the proposed method.

Testcase	Fill Approach	Oxide Density Variation	Minimum Nitride Density	Average Nitride Density
Mixed	Unfilled	11.13%	21.47%	27.56%
	Tiled $0.5\mu/0.5\mu$	11.25%	28.13%	31.89%
	Tiled $1.0\mu/0.5\mu$	12.91%	25.54%	31.25%
	Tiled $1.0\mu/1.0\mu$	12.05%	23.97%	29.59%
	Proposed	2.79%	57.20%	66.34%
OpenRisc8	Unfilled	9.93%	25.87%	36.05%
	Tiled $0.5\mu/0.5\mu$	9.74%	31.91%	38.25%
	Tiled $1.0\mu/0.5\mu$	9.52%	31.50%	38.30%
	Tiled $1.0\mu/1.0\mu$	9.51%	29.02%	37.33%
	Proposed	4.73%	49.61%	59.35%

Table V.2: CMP simulation results for unfilled layout, layout with tiling-based fill insertion, and layout with the proposed fill insertion method. Planarization window is the time window in which polishing can be stopped. Max. final step height is the maximum difference in oxide height after CMP.

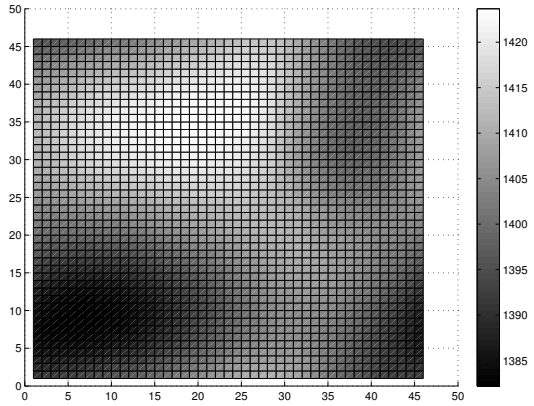
Testcase	Fill Approach	Planarization Window (s)	Max. Final Step Height (nm)
Mixed	Unfilled	45.3	142
	Tiled $0.5\mu/0.5\mu$	46.5	143
	Proposed	53.6	129
OpenRisc8	Unfilled	42.7	146
	Tiled $0.5\mu/0.5\mu$	44.7	144
	Proposed	50.4	133

for variations. Final step height is the difference in oxide thickness after CMP, and is used to quantify oxide dishing. Large final step height leads to poor device characteristics such as excessive leakage and parasitics. Table V.2 presents the planarization window and maximum final step height predictions from the CMP simulator for the unfilled layout, the layout with tiling-based fill, and layout with fill inserted using the proposed methodology. Compared to tiling-based fill, we observe a 17% increase in planarization length and a 9% decrease in maximum final step height on average over the two testcases.

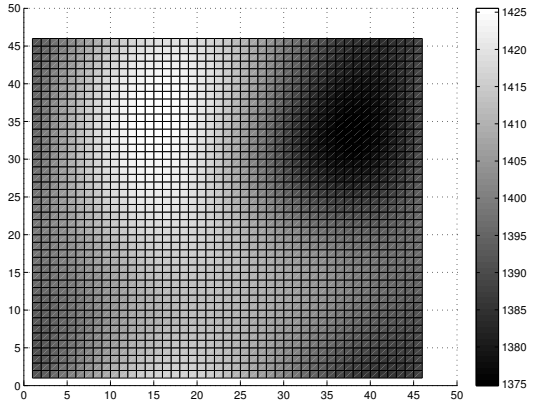
Figure V.12 presents the final step height maps for the the unfilled layout, layout with tiling-based fill, and layout with fill inserted by the proposed methodology. We assume CMP to stop at the middle of the planarization window. The final step height is lower all over the chip when fill is inserted by our approach.

V.C On Capacitive Impact of Floating Fill

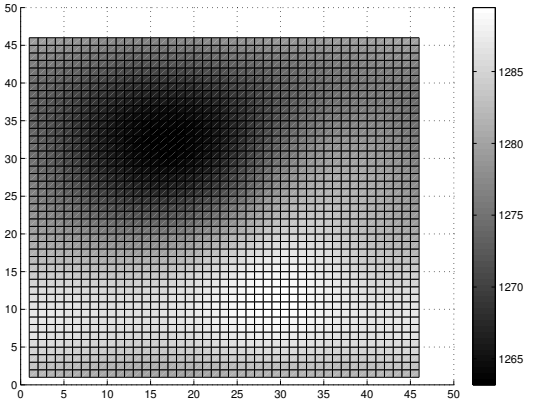
It is well known that BEOL fill insertion can increase the coupling and total interconnect capacitance and consequently deteriorate performance [168, 112].



(a)



(b)



(c)

Figure V.12: Final step height (in angstroms) maps for the unfilled layout (a), layout with tiling-based fill insertion (b), and layout with the proposed insertion method (c).

Traditionally, foundry-supplied design rules have been used by the designers to meet density requirements while not significantly increasing the interconnect capacitance. While fill-insertion design rules have sufficed until now, they are overly conservative and arguably at the end of their life-cycle. Specifically, buffer distance (or keep-off distance) rules have been used to limit the impact of fill on total and coupling capacitance. As crosstalk analysis gains importance and interconnect delay increases, both coupling and total capacitance must be accurately modeled. In the absence of reliable fill extraction tools, buffer distance must be increased. Unfortunately this is not feasible since small density variation may not be achievable if buffer distance is large. Hence, there is a need to relax pessimistic buffer distance rules and explicitly model fill impact on capacitance.

BEOL fill can be grounded by connecting to power or ground nets, or left floating. Floating fill, in comparison to grounded fill, generally offers smaller increase in total capacitance and does not require power/ground routes to the fill geometry. However, floating fill increases coupling capacitance that can lead to signal integrity issues. In the absence of fast and reliable floating fill extraction tools[2], floating fill is cautiously used or not used at all (e.g., in analog circuits). Grounded fill, despite its larger impact on total capacitance and high routing costs that often lead to ECOs, is used as a substitute. Therefore, it is worthwhile to study the impact of floating fill on interconnect capacitance and to develop its trends to aid the designer.

In [142] a model library-based approach to extract floating fill was briefly described. Results demonstrating the accuracy of the approach and characterization time were, however, not presented. [111] presented a methodology for full-chip extraction of *total* capacitance in the presence of floating fill and [112] extended their analysis. Their approach adjusts the permittivity and sidewall thickness of dielectric to account for the capacitance increase due to fill so that off-the-shelf extractors can then be used. In our assessment, quantification of the increase in

[2]Recent full-chip 3D extraction tools support floating fill extraction. Some of these tools, however, implicitly assume regular fill patterns and large buffer distances. Reliable extraction of floating fill arranged in arbitrary patterns is still, to our knowledge, a topic of active research.

dielectric sidewall thickness and permittivity, and identification of regions where the increase must be applied are the main challenges especially when fill insertion is not performed as regular structures. Unfortunately, these details are lacking. In today's context when the buffer distance rules are under $1\mu m$, the need for accurate floating fill extraction is underscored and there is a need for extracting single or small number of fill shapes that may not be arranged regularly between the interconnects.

Previous work has also focused on reducing the capacitance impact of floating fill without explicitly modeling it. [168] presented a methodology to select optimal floating and grounded fill configurations that satisfy a given thickness variation budget and minimize the increase in interconnect capacitance. In that work, very large buffer distances ($> 5\mu m$) were used that are no longer relevant today. Large buffer distances significantly reduce capacitance increase and simplify its estimation. Recently, [81] focused on interconnect design that is aware of the resistive effects of imperfect CMP and capacitance increase due to fill insertion. The paper proposed two useful guidelines – *minimize rows* and *maximize columns* (explained later) – to reduce the floating fill impact on capacitance. In addition, [105] proposed three techniques of fill insertion to reduce interconnect capacitance and the number of fills inserted. It also provided an estimation of the required number of fill geometries for each of the proposed techniques. However, it failed to report the accuracy and reliability of the methods and estimations for densities greater than 30%.

In this section we systematically study the impact of various floating fill configuration parameters such as fill size, fill location, interconnect size, separation from interconnect edges, multiple fill columns and rows, etc. on coupling capacitance. On the basis of our studies, we propose certain guidelines for fill insertion to reduce the capacitance impact of floating fill while achieving the prescribed metal density. Our results indicate significant reduction in coupling capacitance due to fill insertion by using the proposed guidelines.

We study the effect of floating fill on capacitance of same-layer interconnects only. This restriction simplifies our analyses without significantly compro-

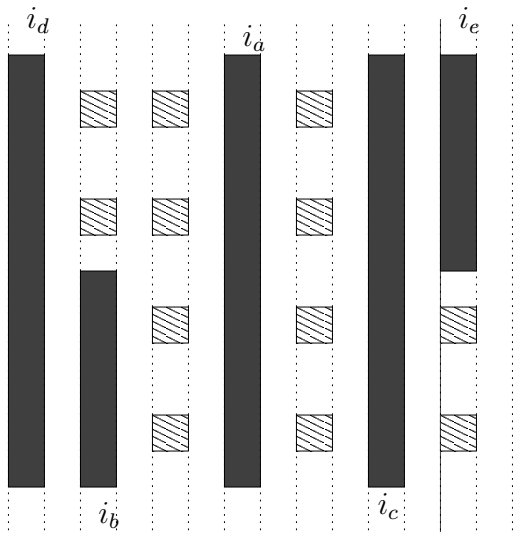


Figure V.13: Assumed Layer M for first set of motivation experiments.

mising the usefulness of our results. We show that: (a) coupling capacitance of an interconnect with same-layer interconnects is a large fraction of its total capacitance, and (b) floating fills on the same layer as two interconnects, significantly increase their coupling capacitance. We perform our experiments on the following three representative interconnect and fill configurations:

1. Medium wire density on layer M, medium wire density on layers $M-1$ and $M+1$.

   We assume that layers $M+1$ and $M-1$ have long parallel wires with area utilization of 33%. Layer M is shown in Figure V.13.

2. Medium wire density on layer M, high wire density on layers $M-1$ and $M+1$.

   We use the same layer M configuration as shown in Figure V.13. Layers $M+1$ and $M-1$ have 50% area utilization with long parallel wires.

3. Low wire density on layer M, low wire density on layers $M-1$ and $M+1$. Layers $M+1$ and $M-1$ have 25% area utilization with long parallel wires and the configuration of layer M is shown in Figure V.14.

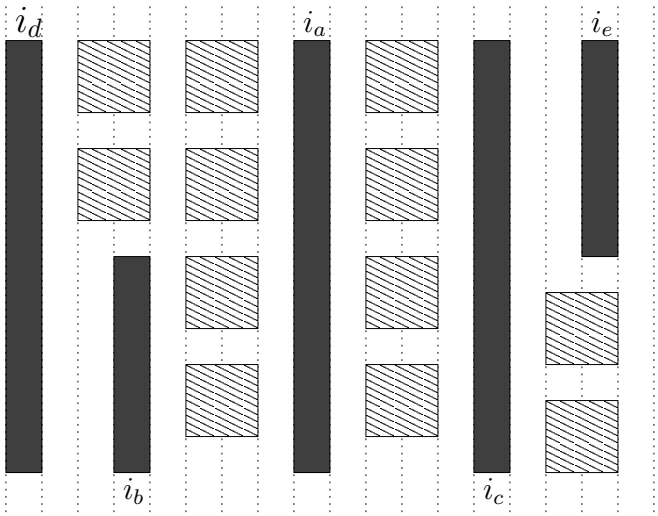


Figure V.14: Assumed Layer M for third set of motivation experiments.

On all layers square fill shapes of side one track are inserted with a minimum separation of one track from wires and other fills. We do not consider high wire density on layer M because no fill can then be inserted. Table V.3 shows the total capacitance and coupling capacitance with same-layer interconnects before and after fill insertion. We use *Synopsys Raphael v-2004.06*, a boundary-element method-based 3D field solver, in all our experiments.

V.C.1 Background

An ideal parallel plate capacitor is a simple geometry that is similar to the configuration formed by two same-layer interconnects. An ideal parallel capacitor is composed of two large parallel metal plates of equal area (A) and small thickness separated by a small distance (d). The parallel-plate capacitance is given by $\epsilon A/d$ where ϵ is the permittivity of the material separating the two plates. Two same-layer interconnects may be viewed as two parallel plates separated by a small distance. However, the height and width of the plates (interconnects) is not large in comparison to the spacing and thus there is a significant divergence from ideal parallel-plate capacitor behavior.

Table V.3: Increase in total capacitance and in same-layer coupling capacitance of interconnect i_a for Figure V.14 on fill insertion.

Deck	Total Capacitance			Same-Layer Coupling		
	Before Fill (fF)	After Fill (fF)	Increase (%)	Before Fill (fF)	After Fill (fF)	Increase (%)
1	0.866	0.955	10.28	0.236	0.312	32.20
2	0.888	0.976	9.91	0.220	0.296	34.55
3	0.828	0.973	17.51	0.141	0.268	90.07

Capacitance of a configuration is directly proportional to the charge accumulated on one of the electrodes ($Q = CV$). The charge density on an electrode depends on the electric field close to the electrode ($E = \sigma/A$). Therefore, the electric field close to an electrode determines the capacitance of a configuration. When a floating plate of thickness t ($t < d$) and the same size as the conductor plates is inserted in the space between the conductors, the capacitance increases to $\epsilon A/(d-t)$ [106]. I.e., the floating plate effectively reduces the distance between the conductors by its thickness. In case of this configuration, electric field lines are uniform in the space between the plates and normal to them. We call this electric field E_{XX} because it begins from a surface that is normal to the X-axis and ends in one that is also normal to the X-axis. Same-layer interconnect pairs with fill geometries inserted between them, however, have two other non-negligible components of electric field: (1) E_{ZZ}, the electric field from top (bottom) of one conductor to top (bottom) of another and to the top (bottom) of fill geometries, and (2) E_{XY}, the electric field from the sidewall of a conductor to the orthogonal sidewall of fill. The different components of electric field are illustrated in Figure V.15. In configurations where these two electric field components are prominent, capacitance behavior with geometry diverges significantly from ideal parallel-plate capacitance with fill.

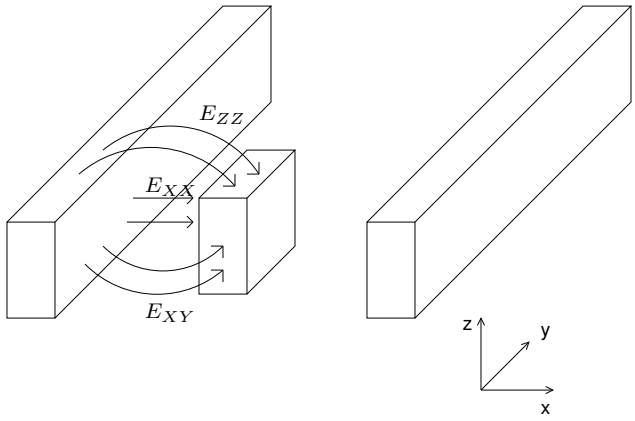


Figure V.15: Different electric field components.

V.C.2 Terminology and Assumptions

We use the following terminology:

- The layer of interest on which the floating fill and coupling interconnects are located is M. $M+1$ and $M-1$ are the layers above and below M respectively. Similarly, $M+2$ and $M-2$ are the layers above $M+1$ and below $M-1$ respectively.
- i_a and i_b are the two wires between which we attempt to study the coupling capacitance C_{ab}. Without loss of generality, we assume that i_a and i_b are vertical.
- R_{ab} is the rectangle enclosed by i_a and i_b on two sides as shown in Figure V.16. If there is no overlap between i_a and i_b in the X-direction, then R_{ab} is undefined. RE_{ab} is R_{ab} extended by the spacing between i_a and i_b on both sides that are orthogonal to i_a and i_b.
- $f_1 \ldots f_n$ are the fill geometries in the region RE_{ab} and the increase in coupling capacitance between i_a and i_b is represented by ΔC_{ab}. We study ΔC_{ab} in all our experiments.
- All sizes are measured in *tracks* and one track is $0.3\mu m$ in keeping with the $90nm$ technology intermediate layer design rules.

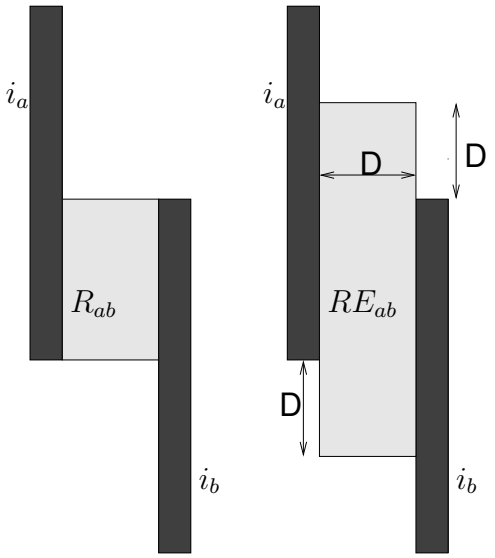


Figure V.16: Rectangle enclosed by interconnects i_a and i_b.

In all our experiments we make the following assumptions:

- The two interconnects i_a and i_b are parallel (i.e., are not doglegs).
- We treat layers M+2 and M-2 as ground planes and validate this assumption in the next section.
- Only $f_1 \ldots f_n$ affect C_{ab} (i.e., only the fill geometries in RE_{ab} must be considered while computing C_{ab}). We validate this assumption in the next section.

V.C.3 Foundations

In this section we present and validate two foundations that, along with the assumptions, reduce the space of possible fill configurations to analyze.

Foundation 1

The coupling capacitance C_{ab} is only affected by the fill geometries lying in the region RE_{ab}. Essentially, we are assuming that the electric field between i_a and i_b is unaffected by fill geometries outside the RE_{ab}. Table V.4 shows the increase in C_{ab} for five configurations. In all five configurations, M of Figure V.17 is used. There is one fill square of side 2 tracks and the location of it is changed

Table V.4: Increase in C_{ab} as a single fill square is moved to the five locations shown in Figure V.17.

Location	ΔC_{ab} (aF)
1	0.0353
2	3.7109
3	13.3265
4	14.6215
5	13.2631

across the five configurations as shown in the figure. Layers M+1 and M-1 are assumed to have 33% density and layers $M-2$ and $M+2$ are assumed to be ground planes (validated in Foundation 2).

Foundation 2

Layers M+2 and M-2 may be treated as ground planes to compute ΔC_{ab} with negligible error. In our validation experiment, i_a and i_b are separated by 4 tracks and both are 18 tracks long. There are three fill squares, each with side 2 tracks, and they are symmetrically placed in RE_{ab}. Layers $M+1$ and $M-1$ have parallel interconnects with area utilization of 33%. ΔC_{ab} is $0.0579fF$ and $0.0520fF$ when the area density on layers $M+2$ and $M-2$ is set to 20% and 33% respectively. The coupling capacitance increase is $0.0447fF$ when layers $M+2$ and $M-2$ are assumed to be ground planes. Most real situations will have higher density than 33% for $M+1$ and $M-1$, which would shield $M+2$ and $M-2$ even more, making their density's impact smaller. Also, the density of $M+2$ and $M-2$ would be higher than 33%. Hence, we expect the error from assuming $M+2$ and $M-2$ as ground planes to be even smaller.

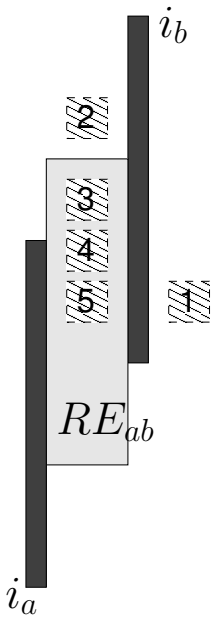


Figure V.17: Five configurations used for Foundation 1 experiments.

V.C.4 Study of Capacitance Impact of Fill

We now present our analyses of the impact of floating fill on coupling capacitance.

Fill Size

Increasing fill *length* (along the interconnects) increases the number of electric field lines between i_a and i_b that get affected. If we ignore the effect due to the electric field lines to/from the horizontal edges of the fill geometry (E_{XY}), a linear increase in ΔC_{ab} with fill length may be expected. Figure V.18(a) shows the relationship between ΔC_{ab} and fill length. The Y-intercept is due to the E_{XY} component of the electric field which is independent of the fill length.

Increasing the fill *width* increases the amount by which electric field lines get affected. In the case of an ideal parallel plate capacitor with plates of area A and separated by distance d, when floating metal of area A and thickness t is inserted, the effective capacitance is given by $kA/(d-t)$ (i.e., the floating metal "eats up" a space that is equal to its thickness from between the two capacitor

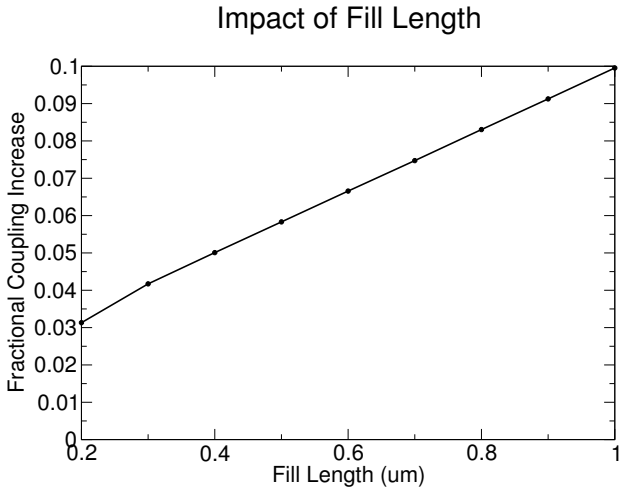
plates). The increase in C_{ab} is therefore expected to be super-linear due to E_{XX}. Figure V.18(b) shows $\Delta C_{ab}/C_{ab}$ when fill width is increased. In our experiments, the spacing between i_a and i_b is fixed at three tracks and the length of wires are set to 17 tracks. Layer $M+1$ and $M-1$ have 33% density while layers $M+2$ and $M-2$ are assumed to be ground planes.

Wire Spacing

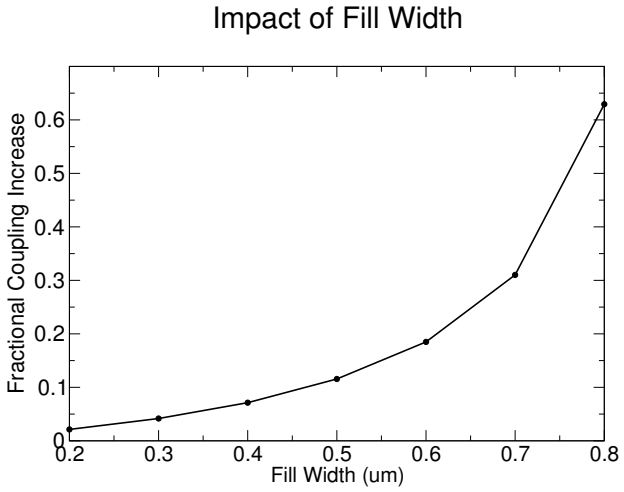
We change the spacing between i_a and i_b and study the impact of floating fill on ΔC_{ab}. Wires i_a and i_b are of length 18 tracks and a fill square of side 2 tracks is placed exactly between the wires and mid-way their length (i.e., at the center of the configuration). The spacing between i_a and i_b is increased from 4 tracks to 10 tracks (fill size is not changed). Layers $M+1$ and $M-1$ have a density of 33%, and layers $M+2$ and $M-2$ are ground planes. Figure V.19(a) shows the impact of spacing on ΔC_{ab}. When the spacing is large (10 tracks), we observe that ΔC_{ab} becomes negative. This likely happens because the fill square starts coupling i_a and i_b to $M+1$ and $M-1$ wires, effectively reducing C_{ab}.

Fill Location

We observe that C_{ab} is unaffected as f_1 is translated along the Y-axis until it starts coming closer to the edge of i_a and/or i_b. We study the edge effects in the next subsection. As the X-coordinate of f_1 is changed, we observe that C_{ab} increases super-linearly. This is likely because as the spacing between f_1 and the nearer wire i_a (without loss of generality) increases, the following two electric fields increase significantly: (1) E_{ZZ}, the electric field between the top (bottom) surface of f_1 and the top (bottom) surface of i_a, and (2) E_{XY}, the electric field between the planes of f_1 to which Y-axis is normal and inside edge of i_a. Figure V.19(b) shows ΔC_{ab} as the spacing between f_1 and i_a is reduced. Wires i_a and i_b are of length 17 tracks and spacing 8 tracks, f_1 is 4 tracks long and 2 tracks wide and its spacing from i_a is reduced from 3 tracks to 0.5 tracks. Layers $M+1$ and $M-1$ have 33% density while layers $M+2$ and $M-2$ are ground planes.

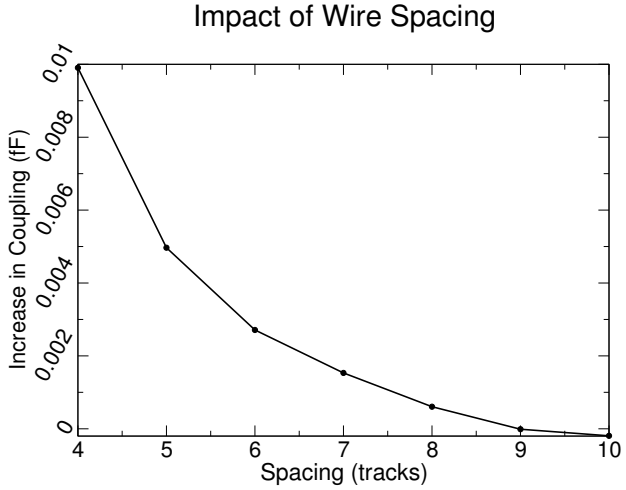


(a)

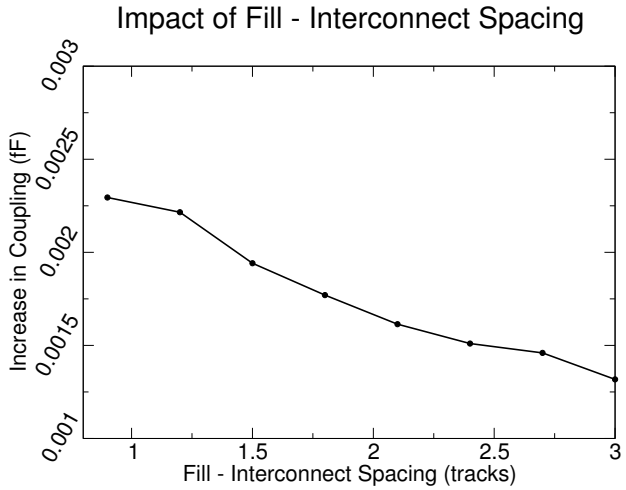


(b)

Figure V.18: Impact of fill size on ΔC_{ab}.



(a)



(b)

Figure V.19: Impact of wire spacing and wire-fill spacing on ΔC_{ab}.

Edge Effects

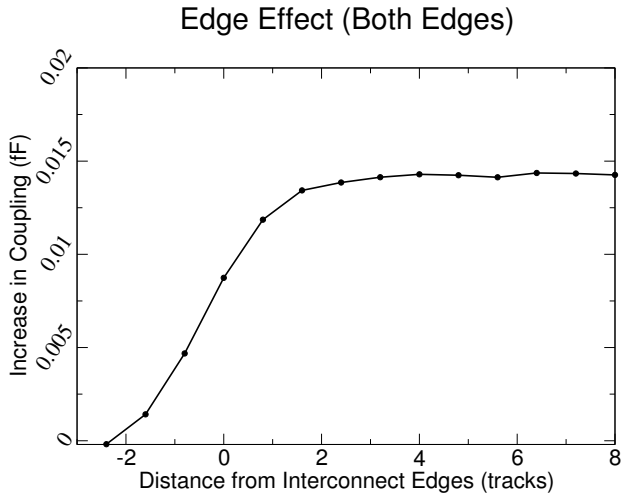
As the fill geometry f_1 is translated in the vertical direction and it approaches the edge(s) of i_a and/or i_b, we observe edge effects. Primarily, E_{XY}, the electric field between the planes of f_1 to which Y-axis is normal and the inside edge of i_a (and/or i_b), reduces as f_1 approaches closer to the edge(s). On further translation, when f_1 is no longer completely in R_{ab}, ΔC_{ab} dramatically decreases. Figures V.20(a) and V.20(b) show ΔC_{ab} as f_1 moves closer to and eventually past the edge(s). In Figure V.20(a), i_a and i_b are horizontally aligned (i.e., their edges are at the same Y-coordinate). In Figure V.20(b) the i_a and i_b are not horizontally aligned and f_1 approaches the edge of i_a while always having i_b on its other side.

Wire Width

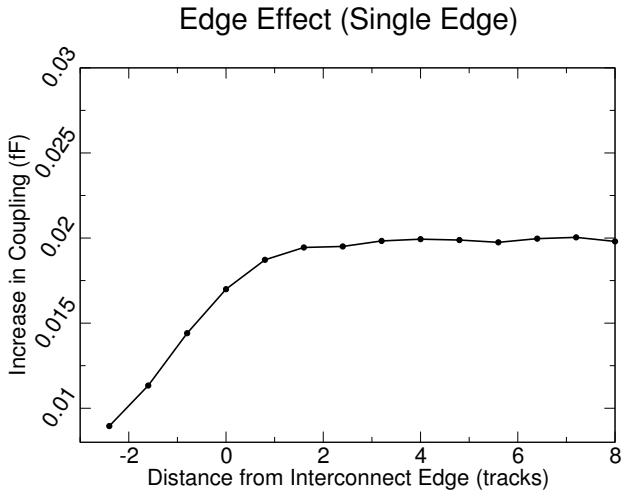
To study the role of wire width, we keep the wire-fill and wire spacing constant and change the width of one wire. We observe that C_{ab} increases as the width surrounding the fill shapes increases. In our experiment, wires i_a and i_b are of length 18 tracks and spaced by 6 tracks, fill shape f_1 is 4 tracks long and 2 tracks wide and has a spacing of 2 tracks with each of the wires. Layers $M+1$ and $M-1$ have 33% density while layers $M+2$ and $M-2$ are assumed to be ground planes. ΔC_{ab} with wire width is shown in Figure V.21 (a). ΔC_{ab} increases rapidly with wire width but saturates at ~ 4 tracks. The electric field component E_{ZZ} likely plays the main role when wire width is increased.

Multiple Columns

Vertically aligned fill geometries are said to be in a *fill column*. We study the impact of increasing the number of fill columns in a fixed width budget. Our simulation results show that as the number of columns between i_a and i_b increases the coupling capacitance (C_{ab}) reduces. Similar results were reported in [168, 81]. Figure V.21(b) illustrates the impact of adding multiple columns on C_{ab}. Wires i_a and i_b are of length 18 tracks and spaced 10 tracks apart. The fill shapes are 2 tracks long by 2 tracks wide. Layers $M+1$ and $M-1$ have 33% density while

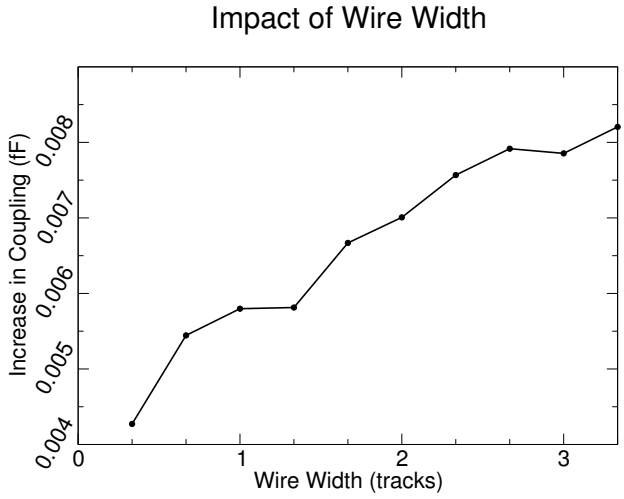


(a)

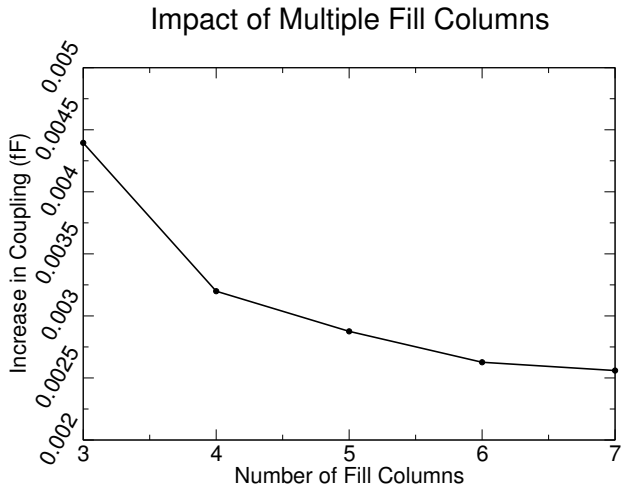


(b)

Figure V.20: Edge effects in computation of ΔC_{ab}.



(a)



(b)

Figure V.21: Impact of wire width and multiple columns on ΔC_{ab}.

layers $M+2$ and $M-2$ are considered as ground planes.

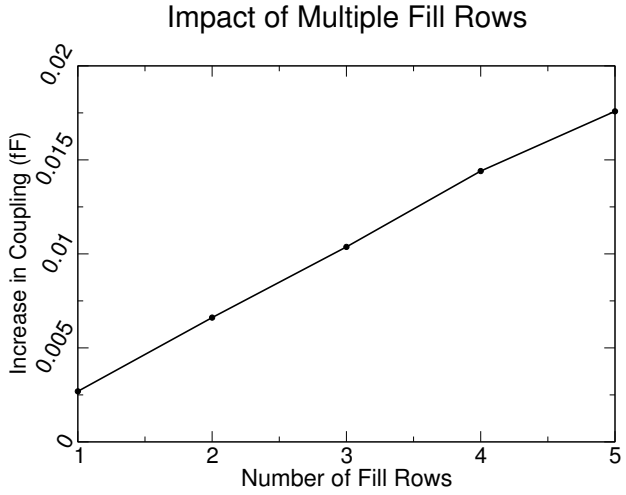
Multiple Rows

In this experiment we increase the number of fill rows which are aligned vertically and observe the impact on the coupling capacitance between i_a and i_b. C_{ab} increases as the number of fill shapes between i_a and i_b increases. As the number of rows is increased, more fill sidewalls are generated that are orthogonal to the interconnect sidewalls and consequently E_{XY} increases. Figure V.22(a) shows the impact of adding fill rows on C_{ab}. We also study the impact of the spacing between two consecutive rows on C_{ab}. Figure V.22(b) illustrates that as spacing between two fill rows decreases, the impact on C_{ab} decreases. We hypothesize that reducing the spacing between two fill rows reduces the total E_{XY}, consequently reducing coupling. Wires i_a and i_b are of length 22 tracks and spaced by 6 tracks. The fill shapes are 2 tracks long and 2 tracks wide. Layers $M+1$ and $M-1$ have 33% density while layers $M+2$ and $M-2$ are assumed to be ground planes.

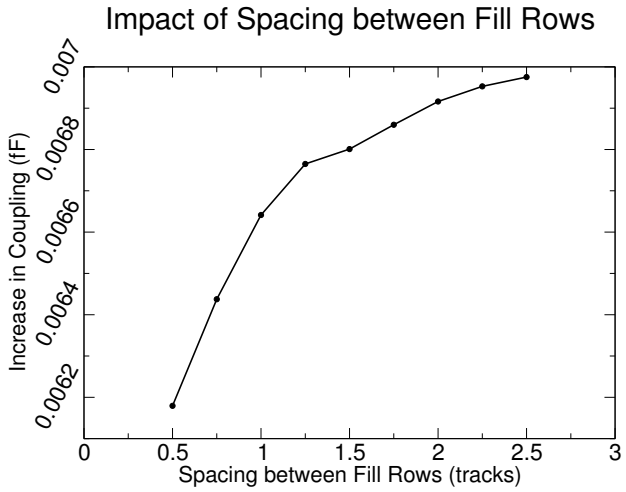
Fill Insertion Guidelines

On the basis of our studies in the previous section we now prescribe certain guidelines for fill insertion between i_a and i_b such that ΔC_{ab} is small. These guidelines may be selectively applied for interconnects that are timing critical or sensitive to noise. We demonstrate in the next section that the increase in coupling capacitance due to floating fill insertion decreases if these guidelines are followed. The guidelines to reduce C_{ab} in order of their decreasing importance are:

1. *High-impact region.* RE_{ab} is the region in which floating fill insertion impacts C_{ab}. Fill insertion must be avoided in R_{ab}.
2. *Edge effects.* Fill insertion should be preferred at the edges of R_{ab} especially in the region $RE_{ab} - R_{ab}$.
3. *Wire spacing.* Impact on C_{ab} is smaller if spacing between i_a and i_b is large hence fill must be inserted where spacing is large.



(a)



(b)

Figure V.22: Impact of multiple rows and consecutive-row spacing on ΔC_{ab}.

4. *Wire width.* Large-width wires are more susceptible to increase in capacitance due to fill. Thinner wire must be preferred as neighbors of fill.
5. *Maximize columns.* The number of columns should be maximized. I.e., fill must be split up subject to the minimum size design rules in a column and spread evenly between i_a and i_b.
6. *Minimize rows.* Fill rows may be merged to reduce C_{ab}.
7. *Increase length not width.* Increasing fill length must be preferred to increasing width to attain the same fill area.
8. *Centralize fill.* Fill or fill configurations when centered between i_a and i_b have a smaller impact on ΔC_{ab}.

V.C.5 Validation

We now validate the guidelines suggested in the previous section. We consider three layer configurations and insert fill first in a regular grid-like fashion and then with our guidelines to attain the same metal density. The *total* capacitance is measured for the two cases with *Raphael*. Layers $M+1$ and $M-1$ have long parallel interconnects with metal density of 33% and layers $M+2$ and $M-2$ are ground planes in all three configurations. We follow these simple design rules:

- Buffer distance is 1 track.
- Minimum fill-to-fill spacing is 1 track.
- Minimum and maximum fill size is 1 and 5 tracks respectively on each side.

Configuration 1

Figure V.23 shows the layer M configuration after fill insertion with floating fill inserted in a regular pattern (in (a)), in a staggered pattern (in (b)), and with our guidelines (in (c)). Layer M contains two interconnects, each of length 22 tracks and width 2 tracks, separated by 11 tracks. Without our guidelines, fill is inserted in a grid to attain a density of $\sim 30\%$ in the region illustrated by the

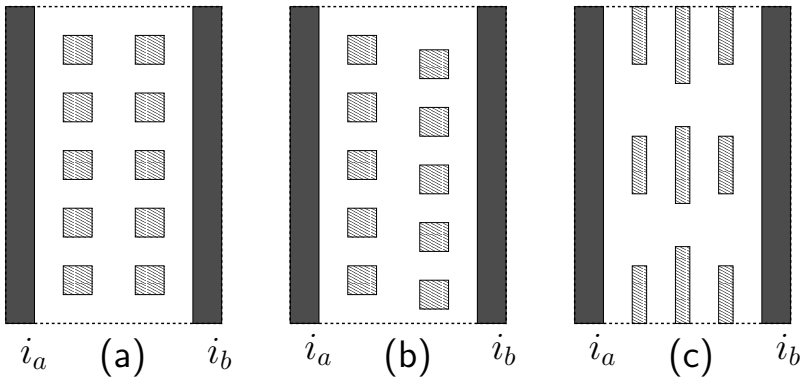


Figure V.23: Configuration 1. (a) regular fill pattern, (b) staggered fill pattern, (c) fill insertion with guidelines.

dashed rectangle.[3] We use Guidelines 2, 5, 6, and 8 to get the configuration shown in Figure V.23 (c). We observe that the coupling capacitance increase with respect to regular (staggered) pattern decreases from 62% (64%) to 16%.

Configuration 2

Figure V.24 shows the layer M configuration after fill insertion is performed in a regular pattern (in (a)), in a staggered pattern (in (b)), and with our guidelines (in (c)). Layer M contains two interconnects, each of length 21 tracks and their width changes from 1 to 2 tracks along their length. Without the guidelines, fill is inserted in a grid-like fashion to attain a density of $\sim 48\%$. Guidelines 4 and 6 are used and we observe that the increase in coupling capacitance with respect to regular (staggered) pattern reduces from 41% (41%) to 30%.

Configuration 3

In this configuration we utilize Guidelines 1, 2, 3, 6, and 8. Layer M with fill inserted in regular and staggered fashion is shown in Figures V.25(a) and V.25(b) respectively. The lengths of Interconnects A, B, C, and D are 18, 27, 27 and 27 tracks respectively and their width is 1 track. Wire B has a dogleg of 2

[3]Tile sizes for density constraints are much larger.

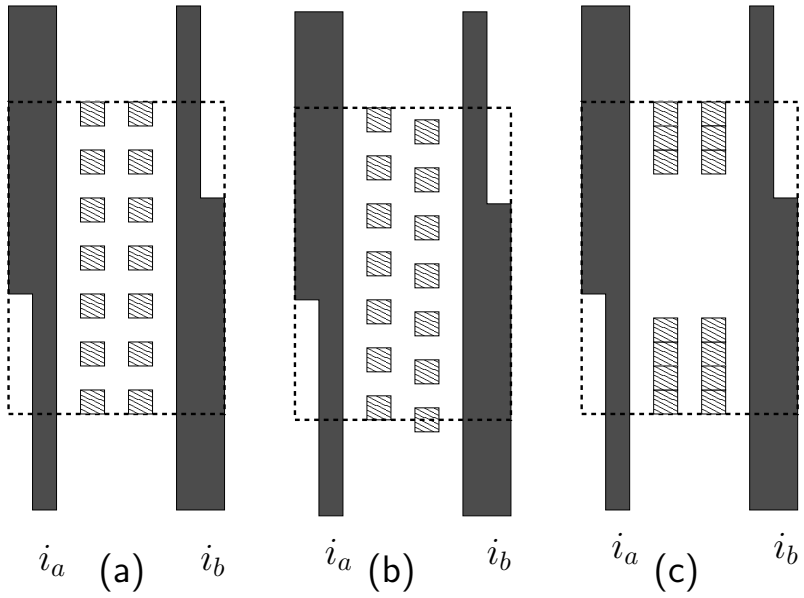


Figure V.24: Configuration 2. (a) regular fill pattern, (b) staggered fill pattern, (c) fill insertion with guidelines.

tracks. Fill insertion is performed to attain a metal density of $\sim 42\%$ in the region illustrated by the dashed rectangle. Figure V.25(c) shows the configuration after fill insertion is performed with our guidelines. Increase in coupling capacitance is 27%, 27%, and 11% when fill is inserted in a regular pattern, staggered pattern, and with our guidelines respectively. Figure V.26 summarizes our results.

V.D Conclusions

CMP is the mainstream planarization technique used in FEOL as well as BEOL to attain stringent planarity requirements. Despite advancements in CMP technology, imperfections exist and are manifested into geometric and electrical variations. Pattern-dependent CMP non-idealities are well-studied: post-CMP topography is dependent on the underlying pattern. Fill insertion is a commonly used method to control feature density and consequently reduce topography variation. In this chapter we presented a fill insertion approach for FEOL that results in superior post-CMP topography. Fill insertion in BEOL causes capacitance of

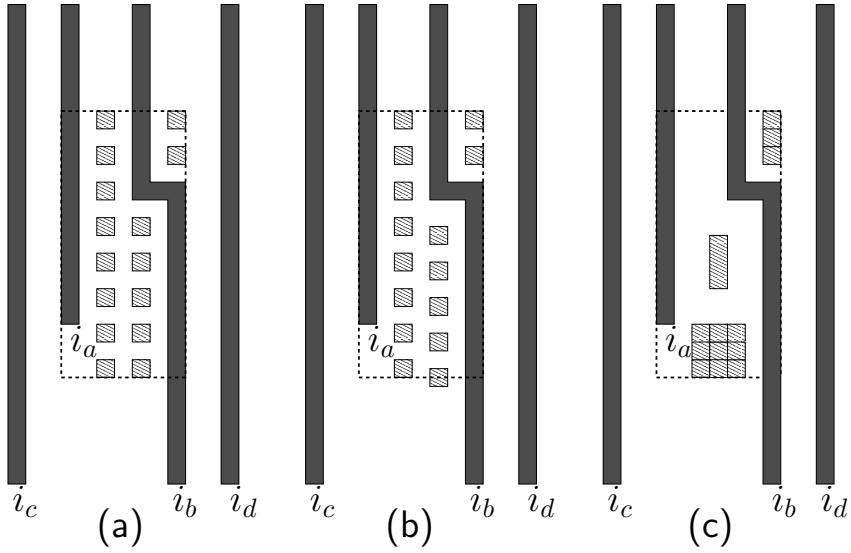


Figure V.25: Configuration 3. (a) regular fill pattern, (b) staggered fill pattern, (c) fill insertion with guidelines.

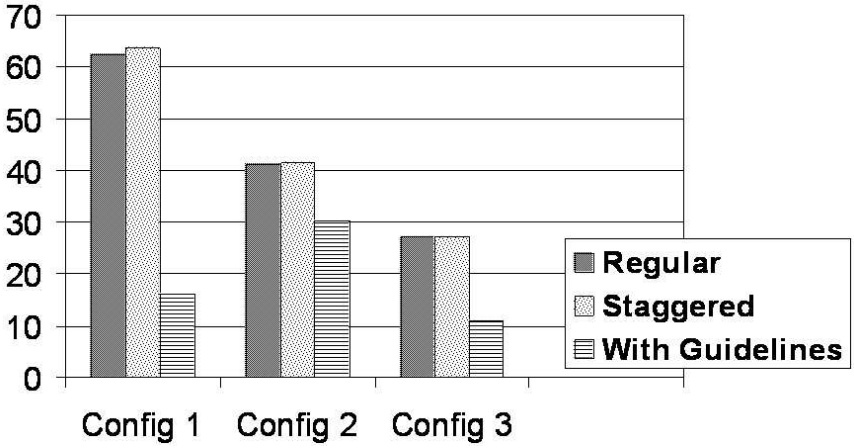


Figure V.26: Percentage increase in coupling capacitance for the three configurations when fill insertion is performed in a regular pattern, in a staggered pattern, or with our guidelines to achieve the same metal density.

nearby wires to increase and the circuit timing is affected. We presented our study of the capacitive effects of BEOL fill and listed guidelines for fill insertion to reduce the capacitive impact.

Our FEOL fill insertion methodology improves STI planarity after CMP. To alleviate the failures caused by imperfect CMP, our approach minimizes oxide density variation and maximizes nitride density. We leverage on the fact that the density of oxide, which is deposited over the nitride, depends on the underlying nitride shapes caused by deposition bias. We first insert maximal fill subject to the design rules and then create holes in it to control the oxide density. Oxide density for minimum density variation is computed with a liner programming-based solution and then nitride is maximized with the computed oxide density as a constraint. To maximize the nitride density, we minimize the number of holes that need to be created. For this, regions that do not contribute to oxide density due to the presence of a hole are approximated by hexagons and an algorithm is proposed to cover the nitride area with the hexagons efficiently. Experimental results indicate a substantial reduction in oxide density variation and increase in nitride density in comparison to traditional tiling-based fill insertion. We also study the post-CMP topography predicted by a CMP simulator for two layouts when fill insertion is done with the proposed method and with traditional tiling-based method. We find the topography of the layouts with our fill insertion to be significantly more desirable than that obtained by traditional tiling-based fill. Specifically, the planarization window increases by 17% and the maximum final step height decreases by 9% on average over our two testcases.

Next, we analyze the increase in coupling capacitance of two same-layer interconnects due to fill insertion on the same layer. We show that same-layer coupling is a large fraction of total capacitance and significantly increases on fill insertion on the same layer. Our studies suggest that the increase in coupling capacitance due to fill insertion:

- is negligible if the fill geometries are outside RE_{ab},
- increases linearly with fill length, but has a Y-intercept due to capacitance

between the interconnect sidewall and the fill plane that is orthogonal to it (E_{xy}),

- increases super-linearly with fill width,
- decreases super-linearly as interconnect-spacing increases,
- decreases approximately super-linearly as the interconnect-fill spacing is increased,
- remains constant as the fill geometry is translated along the length of the interconnects in R_{ab}, then drops sharply in region RE_{ab} and becomes insignificant outside $RE_{ab} - R_{ab}$.
- increases sharply with interconnect width but saturates when the interconnect width becomes ~ 4 tracks,
- decreases as fill area is distributed over multiple columns,
- increases as fill area is distributed over multiple rows, and
- decreases as the spacing between fill rows is reduced.

Based on our observations, we propose eight guidelines to reduce floating fill impact while achieving the prescribed metal density. On our three benchmark configurations, coupling capacitance increase due to floating fill is reduced by $\sim$ 53% on average with the proposed guidelines.

V.E Acknowledgments

This chapter is in part a reprint of:

- A. B. Kahng, P. Sharma and A. Zelikovsky, "Fill for Shallow Trench Isolation CMP," *Proc. International Conference on Computer-Aided Design*, 2006, pp. 661 – 668.

- A. B. Kahng, K. Samadi and P. Sharma, "Study of Floating Fill Impact on Interconnect Capacitance," *Proc. International Symposium on Quality Electronic Design*, 2006, pp. 691 – 696.

I would like to thank my coauthors Kambiz Samadi, Prof. Andrew B. Kahng, and Prof. Alex Zelikovsky.

VI

Conclusions

Increasing variability in today's manufacturing processes causes significant parametric yield loss. DFM refers to measures taken during the design phase to enhance yield. Classical DFM techniques such as use of RETs, design rules and guardbanding, while still necessary, are no longer adequate. The focus of this thesis is on novel DFM techniques that enhance parametric yield.

In this thesis, we classify DFM techniques into four classes based on their approach to parametric yield improvement:

- Systematic variation-aware design analysis and optimization.
- Enhancement of design robustness to process variations.
- Design techniques to reduce process variations.
- Statistical methods.

Utilizing systematic variations in design makes design steps aware of predictable manufacturing non-idealities so that compensations can be made. In Chapters II and III, we presented design analysis and optimization techniques that comprehend systematic variations that arise during exposure in lithography and as a result of mechanical stress in STI. Our defocus-aware leakage estimation and control technique utilizes systematic gate length variations that depend on pitch

and defocus, to improve leakage estimation and optimization. A detailed placement technique optimizes device pitches such that leakage is minimized. We also develop a timing analysis methodology that accounts for systematic gate length variations due to lens aberration. CMOS devices, when subjected to mechanical stress, witness a change in carrier mobility and consequently in delay. Stress arising due to STI is systematic, and we model this phenomenon in timing analysis and placement optimization.

Enhancement of design robustness refers to measures that reduce the sensitivity of design metrics such as delay and leakage to process variations. Such measures reduce the variability of circuit metrics even when process variability is unchanged. Gate length biasing, discussed in Chapter IV, falls into this category. Gate length biasing refers to the selective use of slightly enlarged gate lengths on non-critical paths such that leakage and leakage variability are reduced, even when gate length variability remains the same. We have studied the use of gate length biasing with threshold voltage assignment, along with related manufacturability aspects.

DFM techniques can also increase yield by reducing the variability at the source, i.e., in the process itself. Fill insertion is a mainstream technique used to reduce post-CMP wafer topography variation in both FEOL and BEOL. In Chapter V, we proposed a fill insertion technique for FEOL that reduces topography variation after CMP for STI. The chapter also presents a systematic study of the capacitive effects of BEOL fill insertion on nearby wires. Based on our study, we propose guidelines that reduce capacitive impact of fill while maintaining improved density control. The fourth class of DFM techniques, comprising methods for probabilistic analysis and optimization, was not discussed in this thesis.

Looking into the future, lithography techniques are expected to steadily enhance, but process variability as a percentage of nominal values will continue to increase. Novel DFM methods are needed to control the variability of design metrics and to ensure high parametric yield in the face of high process variability. Unfortunately, acquisition of variational data from manufacturers faces many challenges that must be overcome to enable statistical and systematic-variation

aware techniques. In particular, deployment in fabless design houses will be difficult and delayed because methodologies and standards still need to be developed. Moreover, achieving a design process that is specific to a given manufacturing process may not always be feasible. In the near term, techniques that reduce process variability and that enhance robustness are likely to gain importance. Several techniques in these two categories are already mainstream and will continue to be used (e.g., RETs, fill insertion, and via doubling). Some other techniques such as gate length biasing and use of restricted design rules have gained acceptance and are also becoming part of standard design flows.

Bibliography

[1] "Berkeley Predictive Technology Model," http://www-device.eecs.berkeley.edu/∼ptm/ .

[2] "BSIM4," http://www-device.eecs.berkeley.edu/∼bsim3/bsim4.html .

[3] "Cadence RTL Compiler," http://www.cadence.com/products/digital_ic/rtl_compiler/index.aspx .

[4] "Cadence SignalStorm," http://www.cadence.com/datasheets/6256_LibChar_TP_v2.pdf .

[5] "Cadence SoC Encounter," http://www.cadence.com/products/digital_ic/soc_encounter/index.aspx .

[6] "Mentor Calibre," http://mentor.com/calibre/datasheets/opc/html/ .

[7] "OpenAccess API," http://openeda.si2.org/ .

[8] "OpenCores.org," http://www.opencores.org/projects/ .

[9] "pDfx Design Platform," http://www.pdf.com/services_tech.phtml .

[10] "Prolith," http://www.kla-tencor.com/ .

[11] "Synopsys Design Compiler," http://www.synopsys.com/products/logic/design_compiler.html .

[12] "Synopsys Fammos," http://www.synopsys.com/products/tcad/pa/pa_fammos.html .

[13] "Synopsys HSPICE," http://www.synopsys.com/products/mixedsignal/hspice/hspice.html .

[14] "Synopsys PrimeTime," http://www.synopsys.com/products/analysis/primetime_ds.html .

[15] "Synopsys PrimeTimePX," http://www.synopsys.com/products/solutions/galaxy/power/power.htm .

[16] "Synopsys Seismos," http://www.synopsys.com/products/tcad/pa/pa_seismos.html .

[17] "Synopsys Sentaurus Process," http://www.synopsys.com/products/tcad/sentaurus_procs_ds.html .

[18] A. Agarwal, D. Blaauw, and V. Zolotov, "Statistical Timing Analysis for Intra-Die Process Variations with Spatial Correlations," in *Proc. IEEE International Conference on Computer-Aided Design*, 2003, pp. 900–907.

[19] A. Agarwal, D. Blaauw, V. Zolotov, S. Sundareswaran, M. Zhao, K. Ghala, and R. Panda, "Path-Based Statistical Timing Analysis Considering Inter- and Intra-Die Correlations," in *Proc. ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU)*, 2002, pp. 16–21.

[20] A. Agarwal, C. H. Kim, S. Mukhopadhyay, and K. Roy, "Leakage in Nano-Scale Technologies: Mechanisms, Impact and Design Considerations," in *Proc. ACM/IEEE Design Automation Conference*, 2004, pp. 6–11.

[21] I. Ali, S. Roy, and G. Shinn, "Chemical Mechanical Polishing of Interlayer Dielectric: A Review," in *Solid State Technology*, vol. 37, no. 10, 1994, pp. 63–70.

[22] P. Beckage, T. Brown, R. Tian, E. Travis, A. Phillips, and C. Thomas, "Implementation of Model-Based Tiling at STI CMP for 90nm Technology," in *Proc. Chemical-Mechanical Polish for VLSI/ULSI Multilevel Interconnection Conference*, 2004, pp. 157–162.

[23] P. Beckage, T. Brown, R. Tian, E. Travis, A. Phillips, and C. Thomas, "Prediction and Characterization of STI CMP Within-Die Thickness Variation on 90nm Technology," in *Proc. Chemical-Mechanical Polish for VLSI/ULSI Multilevel Interconnection Conference*, 2004, pp. 267–274.

[24] F. Beeftink, P. Kudva, D. Kung, and L. Stok, "Combinatorial Cell Design for CMOS Libraries," *Integration, the VLSI Journal*, vol. 29, no. 4, pp. 67–93, 2000.

[25] J. L. Bentley, "Experiments on Traveling Salesman Heuristics," in *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 1990, pp. 91–99.

[26] M. Berkelaar, "Statistical Delay Calculation, A Linear Time Model," in *Proc. ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU)*, 1997, pp. 15–24.

[27] F. Boeuf et al., "A Conventional 45nm CMOS Node Low-Cost Platform for General Purpose and Low Power applications," in *Proc. IEEE International Electron Devices Meeting*, 2004, pp. 425–428.

[28] D. Boning and B. Lee, "Nanotopography Issues in Shallow Trench Isolation CMP," in *The Materials Gateway*, 2002, pp. 761–765.

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

[30] A. Bourov, L. C. Litt, and L. Zavyalova, "Impact of Flare on CD Variation for 248nm and 193nm Lithography Systems," in *Proc. SPIE Conference on Optical Microlithography*, 2001, pp. 1388–1393.

[31] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[32] S. P. Boyd, S. J. Kim, D. D. Patil, and M. A. Horowitz, "Digital Circuit Optimization via Geometric Programming," *Operations Research*, vol. 53, no. 6, pp. 899–932, 2005.

[33] E. T. Brist and A. J. Torres, "Model-Assisted Placement of Subresolution Assist Features: Experimental Results," in *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, vol. 5042, 2003, pp. 99–106.

[34] T. Brunner, "Impact of Lens Aberrations on Optical Lithography," in *IBM Journal of Research and Development*, 1997, http://www.research.ibm.com/journal/rd/411/brunner.html.

[35] K. Cao, S. Dobre, and J. Hu, "Standard Cell Characterization Considering Lithography Induced Variations," in *Proc. ACM/IEEE Design Automation Conference*, 2006, pp. 801–804.

[36] Y. Cao, P. Gupta, A. B. Kahng, D. Sylvester, and J. Yang, "Design Sensitivities to Variability: Extrapolations and Assessments in Nanometer VLSI," in *Proc. IEEE International ASIC/SOC Conference*, 2002, pp. 411–415.

[37] Y. Cao, T. Sato, M. Orshansky, D. Sylvester, and C. Hu, "New Paradigm of Predictive MOSFET and Interconnect Modeling for Early circuit Design," in *Proc. IEEE Custom Integrated Circuits Conference*, 2000, pp. 201–204.

[38] C. C. Chang, J. Cong, and M. Xie, "Optimality and Scalability Study of Existing Placement Algorithms," in *Proc. ACM/IEEE Asia and South Pacific Design Automation Conference*, 2003, pp. 621–627.

[39] H. Chang and S. S. Sapatnekar, "Statistical Timing Analysis Considering Spatial Correlations Using a Single PERT-Like Traversal," in *Proc. IEEE International Conference on Computer-Aided Design*, 2003, pp. 621–626.

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

[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. ACM/IEEE Design Automation Conference*, 2005, pp. 71–76.

[42] A. Chatterjee et al., "A Shallow Trench Isolation Study for 0.25/0.18 μm CMOS Technologies and Beyond," in *Proc. IEEE Symposium on VLSI Technology*, 1996, pp. 156–157.

[43] R. S. Chau, "Intel's Breakthrough in High-K Gate Dielectric Drives Moore's Law Well into the Future," in *Technology@Intel Magazine*, 2004.

[44] C.-P. Chen, C. C. N. Chu, and D. F. Wong, "Fast and Exact Simultaneous Gate and Wire Sizing by Lagrangian Relaxation," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 7, pp. 1014–1025, 1999.

[45] P. Chen, D. A. Kirkpatrick, and K. Keutzer, "Miller Factor for Gate-Level Coupling Delay Calculation," in *Proc. IEEE International Conference on Computer-Aided Design*, 2000, pp. 68–75.

[46] Y. Chen, P. Gupta, and A. B. Kahng, "Performance-Impact Limited Area Fill Synthesis," in *Proc. ACM/IEEE Design Automation Conference*, 2003, pp. 22–27.

[47] Y. Chen, A. B. Kahng, G. Robins, and A. Zelikovsky, "Monte-Carlo Algorithms for Layout Density Control," in *Proc. ACM/IEEE Asia and South Pacific Design Automation Conference*, 2000, pp. 523–528.

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

[49] C.-H. Chen et al., "Stress Memorization Technique (SMT) by Selectively Strained-Nitride Capping for Sub-65nm High-Performance Strained-Si Device Application," in *Proc. IEEE Symposium on VLSI Technology*, 2004, pp. 56–57.

[50] M. Cho, D. Z. Pan, H. Xiang, and R. Puri, "Wire Density Driven Global Routing for CMP Variation and Timing," in *Proc. IEEE International Conference on Computer-Aided Design*, 2006, pp. 487–492.

[51] K.-S. Choi et al., "Application of Ceria-Based High-Selectivity Slurry to STI CMP for Sub-0.18um CMOS Technologies," in *Proc. Chemical-Mechanical Polish for VLSI/ULSI Multilevel Interconnection Conference*, 1999, pp. 307–313.

[52] R. Choudhury, *Handbook of Microlithography and Micromachining Volume:1.* SPIE Press Monograph, 2002.

[53] A. Chowdhary, K. Rajagopal, S. Venkatesan, T. Cao, V. Tiourin, Y. Parasuram, and B. Halpin, "How Accurately Can We Model Timing in a Placement Engine?" in *Proc. ACM/IEEE Design Automation Conference*, 2005, pp. 801–806.

[54] W. Chuang, S. Sapatnekar, and I. Hajj, "Delay and Area Optimization for Discrete Gate Sizes under Double-Sided Timing Constraints," in *Proc. IEEE Custom Integrated Circuits Conference*, 1993, pp. 9.4.1–9.4.4.

[55] S.-W. Chung, S.-T. Ahn, H.-C. Sohn, J. Ku, S. Park, Y.-W. Song, H.-S. Park, and S.-D. Lee, "Novel Shallow Trench Isolation Process Using Flowable Oxide CVD for Sub-100nm DRAM," in *Proc. IEEE International Electron Devices Meeting*, 2002, pp. 233–236.

[56] J. Cong, M. Romesis, and M. Xie, "Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms," in *Proc. IEEE International Symposium on Physical Design*, 2003, pp. 88–94.

[57] A. Devgan and C. V. Kashyap, "Block-Based Static Timing Analysis with Uncertainty," in *Proc. IEEE International Conference on Computer-Aided Design*, 2003, pp. 607–614.

[58] K. Doll, F. M. Johannes, and K. J. Antreich, "Iterative Placement Improvement by Network Flow Methods," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 10, pp. 1189–1200, 1994.

[59] L. Economikos et al., "Integrated Electro-Chemical Mechanical Planarization (ECMP) for Future Generation Device Technology," in *Proc. IEEE International Interconnect Technology Conference*, 2004, pp. 233–235.

[60] N. Elbel, Z. Gabric, W. Langheinrich, and B. Neureither, "A New STI Process Based on Selective Oxide Deposition," in *Proc. IEEE Symposium on VLSI Technology*, 1998, pp. 208–209.

[61] N. Farrar, A. Smith, D. Busath, and D. Taitano, "In-Situ Measurement of Lens Aberrations," in *Proc. SPIE Conference on Optical Microlithography*, 2001, pp. 18–29.

[62] J. Fishburn and A. Dunlop, "TILOS: A Posynomial Programming Approach to Transistor-Sizing," in *Proc. IEEE International Conference on Computer-Aided Design*, 1985, pp. 326–328.

[63] D. G. Flagello, H. van der Laan, J. van Schoot, I. Bouchoms, and B. Geh, "Understanding Systematic and Random CD Variations Using Predictive Modelling Techniques," in *Proc. SPIE Conference on Optical Microlithography*, 1999, pp. 162–175.

[64] C. Gallon, G. Reimbold, G. Ghibaudo, R. A. Bianchi, R. Gwoziecki, S. Orain, E. Robilliart, C. Raynaud, and H. Dansas, "Electrical Analysis of Mechanical Stress Induced by STI in Short MOSFETs Using Externally Applied Stress," *IEEE Trans. on Electron Devices*, vol. 51, no. 8, pp. 1254–1261, 2004.

[65] A. Gattiker, S. Nassif, R. Dinakar, and C. Long, "Timing Yield Estimation from Static Timing Analysis," in *Proc. IEEE International Symposium on Quality Electronic Design*, 2001, pp. 437–442.

[66] J. Gortych and D. Williamson, "Effects of Higher-Order Aberrations on the Process Window," in *Proc. SPIE Conference on Optical Microlithography*, 1991, pp. 368–381.

[67] W. Grobman, M. Thompson, R. Wang, C. Yuan, R. Tian, and E. Demircan, "Reticle Enhancement Technology: Implications and Challenges for Physical Design," in *Proc. ACM/IEEE Design Automation Conference*, 2001, pp. 73–78.

[68] P. Groeneveld, "Manufacturability Driven Physical Synthesis," in *IEEE Workshop on Design for Manufacturability and Yield*, 2006, invited talk.

[69] Y. Gu, S. Chang, G. Zhang, K. Kirmse, D. Rogers, and L. Olsen, "Local CD Variation in 65nm Node with PSM Processes STI Topography Characterization (I)," in *Proc. SPIE Conference on Metrology, Inspection and Process Control*, vol. 6152, pp. 29–39, 2006.

[70] P. Gupta and F.-L. Heng, "Toward a Systematic-Variation Aware Timing Methodology," in *Proc. ACM/IEEE Design Automation Conference*, 2004, pp. 321–326.

[71] P. Gupta, A. B. Kahng, Y. Kim, and D. Sylvester, "Self-Compensating Design for Focus Variation," in *Proc. ACM/IEEE Design Automation Conference*, 2005, pp. 365–370.

[72] P. Gupta, A. B. Kahng, Y. Kim, and D. Sylvester, "Self-Compensating Design for Reduction of Timing and Leakage Sensitivity to Systematic Pattern Dependent Variation," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 9, pp. 1614–1624, 2007.

[73] P. Gupta, A. B. Kahng, S. Nakagawa, S. Shah, and P. Sharma, "Lithography Simulation-Based Full-Chip Design Analyses," in *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2006, pp. 61 560T–1 – 61 560T–8.

[74] P. Gupta, A. B. Kahng, and C.-H. Park, "Detailed Placement for Improved Depth of Focus and CD Control," in *Proc. ACM/IEEE Asia and South Pacific Design Automation Conference*, 2005, pp. 343–348.

[75] P. Gupta, A. B. Kahng, and C.-H. Park, "Detailed Placement for Enhanced Control of Resist and Etch CDs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, to appear.

[76] P. Gupta, A. B. Kahng, C.-H. Park, K. Samadi, and X. Xu, "Wafer Topography-Aware Optical Proximity Correction," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 12, pp. 2747–2756, 2006.

[77] P. Gupta, A. B. Kahng, P. Sharma, and D. Sylvester, "Selective Gate-Length Biasing for Cost-Effective Runtime Leakage Control," in *Proc. ACM/IEEE Design Automation Conference*, 2004, pp. 327–330.

[78] L. Hagen, J. H. Huang, and A. B. Kahng, "Quantified Suboptimality of VLSI Layout Heuristics," in *Proc. ACM/IEEE Design Automation Conference*, 1995, pp. 216–221.

[79] J. Halter and F. Najm, "A Gate-level Leakage Power Reduction Method for Ultra Low Power CMOS Circuits," in *Proc. IEEE Custom Integrated Circuits Conference*, 1997, pp. 475–478.

[80] J. P. Han et al., "Novel Enhanced Stressor with Graded Embedded SiGe Source/Drain for High Performance CMOS Devices," in *Proc. IEEE International Electron Devices Meeting*, 2006.

[81] L. He, A. B. Kahng, K. H. Tam, and J. Xiong, "Variability-Driven Considerations in the Design of Integrated-Circuit Global Interconnects," in *Proc.*

International VLSI Multilevel Interconnection Conference, 2004, pp. 214–221.

[82] L. He, A. B. Kahng, K. H. Tam, and J. Xiong, "Simultaneous Buffer Insertion and Wire Sizing Considering Systematic CMP Variation and Random Leff Variation," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 5, pp. 845–857, 2007.

[83] C. Hedlund, H.-O. Blom, and S. Berg, "Microloading Effect in Reactive Ion Etching," *Journal of Vacuum Science and Technology*, vol. 12, no. 4, pp. 1962–1965, 1994.

[84] M. Horiguchi, T. Sakata, and K. Itoh, "Switched-Source-Impedance CMOS Circuit for Low Standby Sub-Threshold Current Giga-Scale LSI's," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 11, pp. 1131–1135, 1993.

[85] S. Hu and J. Hu, "Pattern Sensitive Placement for Manufacturability," in *Proc. IEEE International Symposium on Physical Design*, 2007, pp. 27–34.

[86] I. Hyunsik, T. Inukai, H. Gomyo, T. Hiramoto, and T. Sakurai, "VTC-MOS Characteristics and its Optimum Conditions Predicted by a Compact Analytical Model," in *Proc. IEEE International Symposium on Low Power Electronics and Design*, 2001, pp. 123–128.

[87] T. Iizuka, M. Ikeda, and K. Asada, "Timing-Driven Cell Layout De-Compaction for Yield Optimization," in *Proc. IEEE Design, Automation and Test in Europe*, 2006, pp. 884–889.

[88] W. Jiang, V. Tiwari, E. Iglesia, and A. Sinha, "Topological Analysis for Leakage Prediction of Digital Circuits," in *Proc. ACM/IEEE Asia and South Pacific Design Automation Conference*, 2002, pp. 39–44.

[89] S. Joshi and S. Boyd, "An Efficient Method for Large-Scale Gate Sizing," *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, to appear.

[90] A. B. Kahng, S. Muddu, and P. Sharma, "Defocus-Aware Leakage Estimation and Control," in *Proc. IEEE International Symposium on Low Power Electronics and Design*, 2005, pp. 263–268.

[91] A. B. Kahng, S. Muddu, and P. Sharma, "Detailed Placement for Leakage Reduction Using Systematic Through-Pitch Variation," in *Proc. IEEE International Symposium on Low Power Electronics and Design*, 2007, pp. 110–115.

[92] A. B. Kahng, S. Muddu, and P. Sharma, "Defocus-Aware Leakage Estimation and Control," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, to appear.

[93] A. B. Kahng and C.-H. Park, "Auxiliary Pattern for Cell-Based OPC," in *Proc. SPIE BACUS Symposium on Photomask Technology*, 2006, p. 63494S.

[94] A. B. Kahng, C.-H. Park, P. Sharma, and Q. Wang, "Lens Aberration-Aware Timing-Driven Placement," in *Proc. IEEE Design, Automation and Test in Europe*, 2006, pp. 890–895.

[95] A. B. Kahng and S. Reda, "Evaluation of Placer Suboptimality Via Zero-Change Transformations," in *Proc. IEEE International Symposium on Physical Design*, 2005, pp. 208–215.

[96] A. B. Kahng, G. Robins, H. W. A. Singh, and A. Zelikovsky, "Filling and Slotting: Analysis and Algorithms," in *Proc. IEEE International Symposium on Physical Design*, 1998, pp. 95–102.

[97] A. B. Kahng, G. Robins, A. Singh, and A. Zelikovsky, "Filling Algorithms and Analyses for Layout Density Control," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 4, pp. 445–462, 1999.

[98] A. B. Kahng and K. Samadi, "CMP Fill Synthesis: A Survey of Recent Studies," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, to appear.

[99] A. B. Kahng, K. Samadi, and P. Sharma, "Study of Floating Fill Impact on Interconnect Capacitance," in *Proc. IEEE International Symposium on Quality Electronic Design*, 2006, pp. 691–696.

[100] A. B. Kahng, P. Sharma, and R. O. Topaloglu, "Exploiting STI Stress for Performance," in *Proc. IEEE International Conference on Computer-Aided Design*, 2007, to appear.

[101] A. B. Kahng, P. Sharma, and A. Zelikovsky, "Fill for Shallow Trench Isolation CMP," in *Proc. IEEE International Conference on Computer-Aided Design*, 2006, pp. 661–668.

[102] A. B. Kahng and R. Topaloglu, "A DOE Set for Normalization-Based Extraction of Fill Impact on Capacitances," in *Proc. IEEE International Symposium on Quality Electronic Design*, 2007, pp. 467–474.

[103] J. Kao, S. Narendra, and A. Chandrakasan, "MTCMOS Hierarchical Sizing Based on Mutual Exclusive Discharge Patterns," in *Proc. ACM/IEEE Design Automation Conference*, 1998, pp. 495–500.

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

[105] A. Kurokawa, T. Kanamoto, T. Ibe, A. Kasebe, C. W. Fong, T. Kage, Y. Inoue, and H. Masuda, "Dummy Filling Methods for Reducing Interconnect Capacitance and Number of Fills," in *Proc. IEEE International Symposium on Quality Electronic Design*, 2005, pp. 586–591.

[106] A. Kurokawa, T. Kanamoto, A. Kasebe, Y. Inoue, and H. Masuda, "Efficient Capacitance Extraction Method for Interconnects with Dummy Fills," in *Proc. IEEE Custom Integrated Circuits Conference*, 2004, pp. 485–488.

[107] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization.* Wiley Interscience Series in Pure and Applied Mathematics, 1985.

[108] B. Lee, *Modeling of Chemical Mechanical Polishing for Shallow Trench Isolation*, Ph.D. dissertation, Massachusetts Institute of Technology, 2002.

[109] B. Lee, D. S. Boning, D. L. Hetherington, and D. J. Stein, "Using Smart Dummy Fill and Selective Reverse Etchback for Pattern Density Equalization," in *Proc. Chemical-Mechanical Polish for VLSI/ULSI Multilevel Interconnection Conference*, 2000, pp. 255–258.

[110] D. Lee and D. Blaauw, "Static Leakage Reduction Through Simultaneous Threshold Voltage and State Assignment," in *Proc. ACM/IEEE Design Automation Conference*, 2003, pp. 192–194.

[111] K.-H. Lee, J.-K. Park, Y.-N. Yoon, D.-H. Jung, J.-P. Shin, Y.-K. Park, and J. T. Kong, "Analyzing the Effects of Floating Dummy-Fills: From Feature Scale Analysis to Full-Chip RC Extraction," in *Proc. IEEE International Electron Devices Meeting*, 2001, pp. 685–688.

[112] W.-S. Lee, K.-H. Lee, J.-K. Park, T.-K. Kim, and Y.-K. Park, "Investigation of the Capacitance Deviation due to Metal-Fills and the Effective Interconnect Geometry Modeling," in *Proc. IEEE International Symposium on Quality Electronic Design*, 2003, p. 354.

[113] H. S. Lee et al., "An Optimized Densification of the Filled Oxide for Quarter Micron Shallow Trench Isolation (STI)," in *Proc. IEEE Symposium on VLSI Technology*, 1996, pp. 158–159.

[114] W.-H. Lee et al., "High Performance 65 nm SOI Technology with Enhanced Transistor Strain and Advanced-Low-K BEOL," in *Proc. IEEE International Electron Devices Meeting*, 2005, pp. 61–64.

[115] K.-S. Leung, "SPIDER: Simultaneous Post-Layout IR-Drop and Metal Density Enhancement with Redundant Fill," in *Proc. IEEE International Conference on Computer-Aided Design*, 2005, pp. 33–38.

[116] H. J. Levinson, *Principles of Lithography.* SPIE – The International Society of Optical Engineering, 2001.

[117] 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, no. 9-10, pp. 2117–2120, 2007.

[118] L. W. Liebmann et al., "High-Performance Circuit Design for the RET-Enabled 65-nm Technology Node," in *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2004, pp. 20–29.

[119] J.-J. Liou, K.-T. Cheng, S. Kundu, and A. Krstic, "Fast Statistical Timing Analysis by Probabilistic Event Propagation," in *Proc. ACM/IEEE Design Automation Conference*, 2001, pp. 661–666.

[120] Z. Luo et al., "High Performance and Low Power Transistors Integrated in 65nm Bulk CMOS Technology," in *Proc. IEEE International Electron Devices Meeting*, 2004, pp. 661–664.

[121] C. A. Mack and J. D. Byers, "Improved Model for Focus-Exposure Data Analysis," in *Proc. SPIE Conference on Metrology, Inspection and Process Control*, 2003, pp. 396–405.

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

[123] T. Matsuyama, Y. Shibazaki, Y. Ohmura, and T. Suzuki, "High NA and Low Residual Aberration Projection Lens for DUV Scanner," in *Proc. SPIE Conference on Optical Microlithography*, 2002, pp. 687–695.

[124] J. Mitra, P. Yu, and D. Z. Pan, "RADAR: RET-Aware Detailed Routing Using Fast Lithography Simulations," in *Proc. ACM/IEEE Design Automation Conference*, 2005, pp. 369–372.

[125] 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, 2003.

[126] V. Moroz, G. Eneman, P. Verheyen, F. Nouri, L. Washington, M. Jurczak, D. Pramanik, and X. Xu, "The Impact of Layout on Stress-Enhanced Transistor Performance," in *Proc. IEEE International Conference on Simulaiton of Semiconductor Processes and Devices*, 2005, pp. 143–146.

[127] V. Moroz, L. Smith, X.-W. Lin, D. Pramanik, and G. Rollins, "Stress-Aware Design Methodology," in *Proc. IEEE International Symposium on Quality Electronic Design*, 2006, pp. 807–812.

[128] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V Power Supply High-Speed Digital Circuit Technology with Multithreshold-Voltage CMOS," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, 1995.

[129] S. Mutoh, S. Shigematsu, Y. Matsuya, H. Fukada, T. Kaneko, and J. Yamada, "1V Multithreshold-Voltage CMOS Digital Signal Processor for Mobile Phone Application," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1795–1802, 1996.

[130] Y. Nakahara et al., "A Robust 65-nm Node CMOS Technology for Wide-Range Vdd Operation," in *Proc. IEEE International Electron Devices Meeting*, 2003, pp. 11.2.1–11.2.4.

[131] S. Nakai et al., "A 65 nm CMOS Technology with a High-Performance and Low-Leakage Transistor, a $0.55\mu m^2$ 6T-SRAM Cell and Robust Hybrid-ULK/Cu Interconnects for Mobile Multimedia Applications," in *Proc. IEEE International Electron Devices Meeting*, 2003, pp. 11.3.1–11.3.4.

[132] S. Nojima, S. Mimotogi, M. Itoh, O. Ikenaga, S. Hasebe, K. Hashimoto, S. Inoue, M. Goto, and I. Mori, "Flexible Mask Specifications," in *Proc. SPIE BACUS Symposium on Photomask Technology*, 2002, pp. 187–196.

[133] K. Nose, M. Hirabayashi, H. Kawaguchi, S. Lee, and T. Sakurai, "V_{th} Hopping Scheme to Reduce Subthreshold Leakage for Low-Power Processors," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 3, pp. 413–419, 2002.

[134] M. Orshansky, L. Milor, P. Chen, K. Keutzer, and C. Hu, "Impact of Spatial Intrachip Gate Length Variability on the Performance of High-Speed Digital Circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 5, pp. 544–553, 2002.

[135] 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. 2, pp. 2–11, 2004.

[136] M. Orshansky, L. Milor, L. Nguyen, G. Hill, Y. Peng, and C. Hu, "Intra-Field Gate CD Variability and Its Impact on Circuit Performance," in *Proc. IEEE International Electron Devices Meeting*, 1999, pp. 479–482.

[137] C. Ortolland, "Stress Memorization Technique (SMT) Optimization for 45nm CMOS," in *Proc. IEEE Symposium on VLSI Technology*, 2006, pp. 78–79.

[138] D. Ouma, D. Boning, J. Chung, G. Shinn, L. Olsen, and J. Clark, "An Integrated Characterization and Modeling Methodology for CMP Dielectric Planarization," in *Proc. IEEE International Interconnect Technology Conference*, 1998, pp. 67–69.

[139] Q. Ouyang, M. Ieong, M. Fischetti, S. Panda, D. Boyd, K. Rim, and J. A. Ott, "Characteristics of High Performance PFETs with Embedded SiGe Source/Drain and $< 100 >$ Channels on 45^o Rotated Wafers," in *Proc. IEEE Symposium on VLSI Technology*, 2005, pp. 27–28.

[140] Q. Ouyang et al., "Investigation of CMOS Devices with Embedded SiGe Source/Drain on Hybrid Orientation Substrates," in *Proc. IEEE Symposium on VLSI Technology*, 2005, pp. 28– 29.

[141] J. T. Pan, D. Ouma, P. Li, D. Boning, F. Redecker, J. Chung, and J. Whitby, "Planarization and Integration of Shallow Trench Isolation," in *Proc. International VLSI Multilevel Interconnection Conference*, 1998, pp. 467–472.

[142] J.-K. Park, K.-H. Lee, J.-H. Lee, and Y.-K. Park, "Simulation of Semiconductor Processes and Devices," in *Proc. IEEE International Conference on Simulaiton of Semiconductor Processes and Devices*, 2000, pp. 98–101.

[143] B.-L. Park et al., "Mechanisms of Stress-Induced Voids in Multi-Level Cu Interconnects," in *Proc. IEEE International Interconnect Technology Conference*, 2002, pp. 130–132.

[144] C. V. Peski, "Capabilities of Existing Lithography Tools," in *Proc. International Electronics Manufacturing Technology Symposium*, 1997, pp. 346–348.

[145] S. Postnikove and S. Hector, "ITRS CD Error Budgets: Proposed Simulation Study Methodology," in *International Technology Roadmap for Semiconductors*, 2003.

[146] D. Pramanik, "Impact of Layout on Variability of Devices for Sub 90nm Technologies," in *Electron Devices Society Santa Clara Chapter Meeting*, 2006.

[147] C. Progler, A. Borna, D. Blaauw, and P. Sixta, "Impact of Lithography Cariability on Statistical Timing Behavior," in *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2004, pp. 101–110.

[148] A. Rajaram, J. Hu, and R. Mahapatra, "Reducing Clock Skew Variability via Crosslinks," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 1176–1182, 2006.

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

[150] R. Rao, A. Srivastava, D. Blaauw, and D. Sylvester, "Statistical Analysis of Subthreshold Leakage Current for VLSI Circuits," *IEEE Trans. on Very Large Scale Integrated Systems*, vol. 12, no. 2, pp. 131–139, 2004.

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

[152] P. Royannez et al., "90nm Low Leakage SOC Design Techniques for Wireless Applications," in *Proc. IEEE International Solid-State Circuits Conference*, 2005, pp. 138–140.

[153] S. S. Sapatnekar and W. Chuang, "Power-Delay Optimizations in Gate Sizing," *ACM Transactions on Design Automation of Electronic Systems*, vol. 5, no. 1, pp. 98–114, 2000.

[154] S. S. Sapatnekar, V. Rao, P. Vaidya, and S. Kang, "An Exact Solution to the Transistor Sizing Problem for CMOS Circuits Using Convex Optimization," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 11, pp. 1621–1634, 1993.

[155] D. K. Schroder and J. A. Babcock, "Negative Bias Temperature Instability: Road to Cross in Deep Submicron Silicon Semiconductor Manufacturing," *Journal of Applied Physics*, vol. 94, no. 1, pp. 1–18, 2003.

[156] J. R. Sheats and B. W. Smith, *Mircolithography: Science and Technology*. Marcell Dekker, 1998.

[157] S. Shigematsu, S. Mutoh, Y. Matsuya, Y. Tabae, and J. Yamada, "A 1-V High-Speed MTCMOS Circuit Scheme for Power-Down Application Circuits," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 6, pp. 861–869, 1997.

[158] Y. Shiode, S. Okada, H. Takamori, H. Matusda, and S. Fujiwara, "Method of Zernike Coefficients Extraction for Optics Aberration Measurement," in *Proc. SPIE Conference on Optical Microlithography*, 2002, pp. 138–147.

[159] J. Singh and S. S. Sapatnekar, "Statistical Timing Analysis with Correlated Non-Gaussian Parameters Using Independent Component Analysis," in *Proc. ACM/IEEE Design Automation Conference*, 2006, pp. 155–160.

[160] A. Singhee, C. F. Fang, J. D. Ma, and R. A. Rutenbar, "Probabilistic Interval-Valued Computation: Toward a Practical Surrogate for Statistics Inside CAD Tools," in *Proc. ACM/IEEE Design Automation Conference*, 2006, pp. 167–172.

[161] S. Sirichotiyakul, T. Edwards, C. Oh, R. Panda, and D. Blaauw, "Duet: An Accurate Leakage Estimation and Optimization Tool for Dual-V_{th} Circuits," *IEEE Trans. on Very Large Scale Integrated Systems*, vol. 10, pp. 79–90, 2002.

[162] S. Sirichotiyakul, T. Edwards, C. Oh, J. Zuo, A. Dharchoudhury, R. Panda, and D. Blaauw, "Stand-by Power Minimization through Simultaneous Threshold Voltage Selection and Circuit Sizing," in *Proc. ACM/IEEE Design Automation Conference*, 1999, pp. 436–441.

[163] N. Sirisantana, L. Wei, and K. Roy, "High-Performance Low-Power CMOS Circuits Using Multiple Channel Length and Multiple Oxide Thickness," in *Proc. IEEE International Conference on Computer Design*, 2000, pp. 227–232.

[164] C. S. Smith, "Piezoresistance Effect in Germanium and Silicon," *Physical Review*, vol. 94, no. 1, pp. 42–49, 1953.

[165] A. Srivastava and D. Sylvester, "Minimizing Total Power by Simultaneous V_{dd}/V_{th} Assignment," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 5, pp. 665–677, 2004.

[166] A. Srivastava, D. Sylvester, and D. Blaauw, "Power Minimization Using Simultaneous Gate Sizing, Dual-Vdd and Dual-Vth Assignment," in *Proc. ACM/IEEE Design Automation Conference*, 2004, pp. 783–787.

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

[168] B. Stine, D. Boning, J. Chung, and L. Camilletti, "The Physical and Electrical Effects of Metal-Fill Patterning Practices for Oxide Chemical-Mechanical

Polishing Processes," in *IEEE Trans. on Electron Devices*, vol. 45, 1998, pp. 665–679.

[169] B. Stine, D. Ouma, R. Divecha, D. S. Boning, J. Chung, D. Hertherington, C. R. Harwood, O. S. Nakagawa, and S.-Y. Oh, "Rapid Characterization and Modeling of Pattern Dependent Variation in Chemical-Mechanical Polishing," in *IEEE Trans. on Semiconductor Manufacturing*, vol. 11, 1998, pp. 129–140.

[170] Y. Tateshita et al., "High-Performance and Low-Power CMOS Device Technologies Featuring Metal/High-k Gate Stacks with Uniaxial Strained Silicon Channels on (100) and (110) Substrates," in *Proc. IEEE International Electron Devices Meeting*, 2006.

[171] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices.* Cambridge University Press, 1998.

[172] H. Tennakoon and C. Sechen, "Gate Sizing Using Lagrangian Relaxation Combined With a Fast Gradient-Based Pre-Processing Step," in *Proc. IEEE International Conference on Computer-Aided Design*, 2002, pp. 395–402.

[173] S. E. Thompson, "Strained Si and the Future Direction of CMOS," in *Proc. International Workshop on System-on-Chip for Real-Time Applications*, 2005, pp. 14–16.

[174] R. Tian, D. F. Wong, and R. Boone, "Model-Based Dummy Feature Placement for Oxide Chemical Mechanical Polishing Manufacturability," in *Proc. ACM/IEEE Design Automation Conference*, 2000, pp. 667–670.

[175] K. K. H. Toh and A. Neureuther, "Identifying and Monitoring Effects of Lens Aberrations in Projection Printing," in *Proc. SPIE Conference on Optical Microlithography*, 1987, pp. 202–209.

[176] R. O. Topaloglu, "Energy-Minimization Model for Fill Synthesis," in *Proc. IEEE International Symposium on Quality Electronic Design*, 2007, pp. 444–451.

[177] H. Tsuno et al., "Advanced Analysis and Modeling of MOSFET Characteristic Fluctuation Caused by Layout Variation," in *Proc. IEEE Symposium on VLSI Technology*, 2007, pp. 204–205.

[178] T. Tugbawa, *Chip-Scale Modeling of Pattern Dependencies in Copper Chemical Mechanical Polishing Processes*, Ph.D. dissertation, Massachusetts Institute of Technology, 2002.

[179] G. Venkataraman et al., "Practical Techniques to Reduce Skew and Its Variations in Buffered Clock Networks," in *Proc. IEEE International Conference on Computer-Aided Design*, 2005, pp. 592–596.

[180] C. Visweswariah, "Death, Taxes and Failing Chips," in *Proc. ACM/IEEE Design Automation Conference*, 2003, pp. 343–347.

[181] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, "First-Order Incremental Block-Based Statistical Timing Analysis," in *Proc. ACM/IEEE Design Automation Conference*, 2004, pp. 331–336.

[182] I. A. Wagner and I. Koren, "An Interactive VLSI CAD Tool for Yield Estimation," *IEEE Trans. on Semiconductor Manufacturing*, vol. 8, no. 2, 1995.

[183] W.-S. Wang, V. Kreinovich, and M. Orshansky, "Statistical Timing Based on Incomplete Probabilistic Descriptions of Parameter Uncertainty," in *Proc. ACM/IEEE Design Automation Conference*, 2006, pp. 161–166.

[184] L. Wei, Z. Chen, M. Johnson, K. Roy, and V. De, "Design and Optimization of Low Voltage High Performance Dual Threshold CMOS Circuits," in *Proc. ACM/IEEE Design Automation Conference*, 1998, pp. 489–494.

[185] L. Wei, Z. Chen, K. Roy, Y. Ye, and V. De, "Mixed-V_{th} CMOS Circuit Design Methodology for Low Power Applications," in *Proc. ACM/IEEE Design Automation Conference*, 1999, pp. 430–435.

[186] L. Wei, K. Roy, and C. K. Koh, "Power Minimization by Simultaneous Dual-V_{th} Assignment and Gate-Sizing," in *Proc. ACM/IEEE Design Automation Conference*, 2000, pp. 413–416.

[187] A. K. Wong, "FastPlace: Efficient Analytical Placement Using Cell Shifting, Iterative Local Refinement and a Hybrid Net Model," in *Proc. SPIE Conference on Optical Microlithography*, 2002, pp. 395–368.

[188] A. K. Wong, R. A. Ferguson, L. W. Liebmann, S. M. Mansfield, A. F. Molless, and M. O. Neisser, "Lithographic Effects of Mask Critical Dimension Error," in *Proc. SPIE Conference on Optical Microlithography*, 1998, pp. 106–116.

[189] A. K.-K. Wong, *Resolution Enhancement Techniques in Optical Lithography.* SPIE – The International Society of Optical Engineering, 2001.

[190] H. Xiang, L. Deng, R. Puri, K.-Y. Chao, and M. D. F. Wong, "Dummy Fill Density Analysis with Coupling Constraints," in *Proc. IEEE International Symposium on Physical Design*, 2007, pp. 3–9.

[191] X. Xie, T. Park, D. Boning, A. Smith, P. Allard, and N. Patel, "Characterizing STI CMP Processes with an STI Test Mask Having Realistic Geometric Shapes," in *Proc. Chemical-Mechanical Polishing Symposium, MRS Spring Meeting*, 2004.

[192] J. Yang, L. Capodieci, and D. Sylvester, "Advanced Timing Analysis Based on Post-OPC Extraction of Critical Dimensions," in *Proc. ACM/IEEE Design Automation Conference*, 2005, pp. 359–364.

[193] H. S. Yang et al., "Dual Stress Liner for High Performance Sub-45nm Gate Length SOI CMOS Manufacturing," in *Proc. IEEE International Electron Devices Meeting*, 2004, pp. 1075–1077.

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

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

[196] Y. Zhan, A. J. Strojwas, X. Li, L. T. Pileggi, D. Newmark, and M. Sharma, "Correlation-Aware Statistical Timing Analysis with Non-Gaussian Delay Distributions," in *Proc. ACM/IEEE Design Automation Conference*, 2005, pp. 77–82.

[197] L. Zhang, W. Chen, Y. Hu, J. A. Gubner, and C. C.-P. Chen, "Correlation-Preserved Non-Gaussian Statistical Timing Analysis with Quadratic Timing Model," in *Proc. ACM/IEEE Design Automation Conference*, 2005, pp. 83–88.

[198] Y. Zhou, Z. Li, Y. Tian, W. Shi, and F. Liu, "A New Methodology for Interconnect Parasitics Extraction Considering Photo-Lithography Effects," in *Proc. ACM/IEEE Asia and South Pacific Design Automation Conference*, 2007, to appear.