UNIVERSITY OF CALIFORNIA, SAN DIEGO

Mitigation of Variability and Reliability Margins in IC Implementation

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Computer Engineering)

by

Tuck Boon Chan

Committee in charge:

Professor Andrew B. Kahng, Chair
Professor Chung-Kuan Cheng
Professor Puneet Gupta
Professor Rajesh Gupta
Professor Bill Lin
Professor Jason Schweinsberg

2014

The dissertation of Tuck Boon Chan is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2014

DEDICATION

I dedicate this thesis to my loving wife, Yu-Ying Huang. Without her encouragement and sacrifice this thesis would not have been finished.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Foremost, I would like to thank my parents (my father Ee Fah Chan and my mother Lim Lim Mui), my wife Yu-Ying Huang, her parents (her father Kuo-Er Huang and her mother Huan-Jen Liu) and my sister for their endless love throughout the years. Their patience and support have been the most important contributions to my course.

I would like to thank my advisor Professor Andrew B. Kahng for his invaluable advice and continuous support. His attitude and principles on research have provided a precious lesson throughout my Ph.D. study. His immense knowledge and guidance helped me in conducting research and in the writing of this thesis.

Besides my advisor, I would like to thank my thesis committee members, Professor Chung-Kuan Cheng, Professor Puneet Gupta, Professor Rajesh Gupta, Professor Bill Lin, and Professor Jason Schweinsberg, for their time to review my research and for their valuable comments. Especially, I thank Professor Puneet Gupta for his advice and encouragement at the beginning of my Ph.D. studies.

I thank Mr. Sorin Dobre for mentoring me on many of my research projects and offering me the summer internship.

Last, but not least, I would like to thank my fellow labmates in the UCSD VLSI CAD Laboratory (Siddhartha Nath, Vaishnav Srinivas, Wei-Ting Chan, Jiajia Li, Hyein Lee, Kwangsoo Han) and former lab members (Dr. Kwangok Jeong, Professor Seokhyeong Kang, Jingwei Lu and Ilgweon Kang) for the stimulating discussions, the sleepless nights, and the many great times together.

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

- Chapter 2 is based on the following publications.
  - **Tuck-Boon Chan**, Puneet Gupta, Andrew B. Kahng and Liangzhen Lai, "Synthesis and Analysis of Design-Dependent Ring Oscillator (DDRO) Performance Monitors", *IEEE Transactions on Very Large Scale Integration Systems*, to appear.
  - **Tuck-Boon Chan**, Puneet Gupta, Andrew B. Kahng and Liangzhen Lai, "DDRO: A Novel Performance Monitoring Methodology Based on Design-Dependent Ring Oscillators", *Proc. International Symposium on Quality Electronic Design*, 2012, pp. 633-640.

  - **Tuck-Boon Chan** and Andrew B. Kahng, "Tunable Sensors for Process-Aware Voltage Scaling", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2012, pp. 7-14.
  - **Tuck-Boon Chan** and Andrew B. Kahng, "Post-Routing Back-End-of-the-Line Layout Optimization for Improved Time-Dependent Dielectric Breakdown Reliability", *Proc. SPIE Conference on Design for Manufacturability through Design-Process Integration*, 2013, pp. 86840L-1-86840L-11.

- Chapter 3 is based on the following publications.

  - **Tuck-Boon Chan**, Wei-Ting Jonas Chan and Andrew B. Kahng, "On Aging-Aware Signoff with Adaptive Voltage", *IEEE Transactions On Circuits and Systems* I, to appear.
  - **Tuck-Boon Chan**, Wei-Ting Jonas Chan and Andrew B. Kahng, "Impact of Adaptive Voltage Scaling on Aging-Aware Signoff", *Proc. Design, Automation and Test in Europe*, 2013, pp. 1683-1688.
  - **Tuck-Boon Chan**, Andrew B. Kahng, Jiajia Li and Siddhartha Nath, "Optimization of Overdrive Signoff", *Proc. Asia and South Pacific Design Automation Conference*, 2013, pp. 344-349.
  - **Tuck-Boon Chan**, Andrew B. Kahng, Jiajia Li, Siddhartha Nath and Bong-Il Park "Optimization of Overdrive Signoff in High-Performance and Low-Power ICs", *IEEE Transactions on Very Large Scale Integration Systems*, to appear.
  - **Tuck-Boon Chan**, Sorin Dobre and Andrew B. Kahng, "Timing Signoff Using Tightened Back-End-of-Line Corners in Advanced Technology Nodes", *Proc. IEEE International Conference on Computer Design*, to appear.

- Chapter 4 is based on the following publications.

  - **Tuck-Boon Chan** and Puneet Gupta, "On Electrical Modeling of Imperfect Diffusion Patterning", *Proc. IEEE/ACM International Conference on VLSI Design*, 2010, pp. 224-229.
  - **Tuck-Boon Chan**, Kwangok Jeong and Andrew B. Kahng, "Performance and Variability Driven Guidelines for BEOL Layout Decomposition with LELE Double Patterning", *Proc. SPIE/BACUS Symposium on Photomask Technology and Management*, 2011, pp. 81663O-1-81663O-12.

- **Tuck-Boon Chan**, Abde Ali Kagalwalla and Puneet Gupta, "Measurement and Optimization of Electrical Process Window", *SPIE Journal of Microlithography, Microfabrication and Microsystems* 10(1) (2011), pp. 013014-1-013014-14.
- **Tuck-Boon Chan**, Abde Ali Kagalwalla and Puneet Gupta, "Measurement and Optimization of Electrical Process Window", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2010, pp. 76410J-1-76410J-11.
- **Tuck-Boon Chan**, Aashish Pant, Lerong Cheng and Puneet Gupta, "Design-Dependent Process Monitoring for Wafer Manufacturing and Test Cost Reduction", *IEEE Transactions on Semiconductor Manufacturing* 25(3) (2012), pp. 447-459.
- **Tuck-Boon Chan**, Aashish Pant, Lerong Cheng and Puneet Gupta, "Design-Dependent Process Monitoring for Back-End Manufacturing Cost Reduction", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2010, pp. 116-122.

My co-authors (Wei-Ting Jonas Chan, Lerong Cheng, Sorin Dobre, Professor Puneet Gupta, Dr. Kwangok Jeong, Abde Ali Kagalwalla, Professor Andrew B. Kahng, Liangzhen Lai, Jiajia Li, Siddhartha Nath, Aashish Pant, and Dr. Bong-Il Park, listed in alphabetical order) have all kindly approved the inclusion of the aforementioned publications in my thesis.

VITA

1980	Born, Kuala Lumpur, Malaysia
2003	B.Sc., Electrical Engineering, University of Technology, Johor Bahru, Malaysia
2003	Engineer, HPI Innovation Sdn. Bhd., Johor Bahru, Malaysia
2007	M.Sc., Electrical Engineering, National Taiwan University, Taipei, Taiwan
2014	C.Phil., Electrical Engineering (Computer Engineering), University of California, San Diego
2014	Ph.D., Electrical Engineering (Computer Engineering), University of California, San Diego

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

- **Tuck-Boon Chan**, Andrew B. Kahng, Jiajia Li, Siddhartha Nath and Bong-Il Park, "Optimization of Overdrive Signoff in High-Performance and Low-Power ICs", *IEEE Transactions on Very Large Scale Integration Systems*, to appear.
- **Tuck-Boon Chan**, Wei-Ting J. Chan and Andrew B. Kahng, "On Aging-Aware Signoff with Adaptive Voltage", *IEEE Transactions On Circuits and Systems* I, to appear.
- **Tuck-Boon Chan**, Puneet Gupta, Andrew B. Kahng and Liangzhen Lai, "Synthesis and Analysis of Design-Dependent Ring Oscillator (DDRO) Performance Monitors", *IEEE Transactions on Very Large Scale Integration Systems*, to appear.
- **Tuck-Boon Chan**, Aashish Pant, Lerong Cheng and Puneet Gupta, "Design-Dependent Process Monitoring for Wafer Manufacturing and Test Cost Reduction", *IEEE Transactions on Semiconductor Manufacturing* 25(3) (2012), pp. 447-459.
- **Tuck-Boon Chan**, Abde Ali Kagalwalla and Puneet Gupta, "Measurement and Optimization of Electrical Process Window", *SPIE Journal of Microlithography, Microfabrication and Microsystems* 10(1) (2011), pp. 013014-1-013014-14.
- **Tuck-Boon Chan**, Sorin Dobre and Andrew B. Kahng, "Timing Signoff Using Tightened Back-End-of-Line Corners in Advanced Technology Nodes", *Proc. IEEE International Conference on Computer Design*, to appear.

- **Tuck-Boon Chan**, Puneet Gupta, Kwangsoo Han, Abde Ali Kagalwalla, Andrew B. Kahng and Emile Sahouria, "Benchmarking of Mask Fracturing Heuristics", *Proc. IEEE/ ACM International Conference on Computer-Aided Design*, 2014, to appear.

- **Tuck-Boon Chan**, Kwangsoo Han, Andrew B. Kahng, Jae-Gon Lee and Siddhartha Nath, "OCV-Aware Top-Level Clock Tree Optimization", *Proc. Great Lakes Symposium on Very Large Scale Integration*, 2014, pp. 33-38.

- **Tuck-Boon Chan**, Andrew B. Kahng and Jiajia Li, "NOLO: A No-Loop, Predictive Useful Skew Methodology for Improved Timing in IC Implementation", *Proc. International Symposium on Quality Electronic Design*, 2014, pp. 504-509.

- **Tuck-Boon Chan**, Andrew B. Kahng and Jiajia Li, "Toward Quantifying the IC Design Value of Interconnect Technology Improvements", *Proc. ACM International Workshop on System-Level Interconnect Prediction*, 2013, pp. 1-6.

- **Tuck-Boon Chan**, Andrew B. Kahng and Jiajia Li, "Reliability-Constrained Die Stacking Order in 3DICs under Manufacturing Variability", *Proc. International Symposium on Quality Electronic Design*, 2013, pp. 16-23.

- **Tuck-Boon Chan**, Wei-Ting J. Chan and Andrew B. Kahng, "Impact of Adaptive Voltage Scaling on Aging-Aware Signoff", *Proc. Design, Automation and Test in Europe*, 2013, pp. 1683-1688.

- **Tuck-Boon Chan** and Andrew B. Kahng, "Post-Routing Back-End-of-the-Line Layout Optimization for Improved Time-Dependent Dielectric Breakdown Reliability", *Proc. SPIE Conference on Design for Manufacturability through Design-Process Integration*, 2013, pp. 86840L-1-86840L-11.

- **Tuck-Boon Chan**, Andrew B. Kahng, Jiajia Li and Siddhartha Nath, "Optimization of Overdrive Signoff", *Proc. Asia and South Pacific Design Automation Conference*, 2013, pp. 344-349.

- **Tuck-Boon Chan** and Andrew B. Kahng, "Tunable Sensors for Process-Aware Voltage Scaling", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2012, pp. 7-14.

- **Tuck-Boon Chan** and Andrew B. Kahng, "Improved Path Clustering for Adaptive Path-Delay Testing", *Proc. International Symposium on Quality Electronic Design*, 2012, pp. 13-20.

- **Tuck-Boon Chan**, Puneet Gupta, A. Kahng and Liangzhen Lai, "DDRO: A Novel Performance Monitoring Methodology Based on Design-Dependent Ring Oscillators", *Proc. International Symposium on Quality Electronic Design*, 2012, pp. 633-640.
- **Tuck-Boon Chan**, K. Jeong and Andrew B. Kahng, "Performance and Variability Driven Guidelines for BEOL Layout Decomposition with LELE Double Patterning", *Proc. SPIE/ BACUS Symposium on Photomask Technology and Management*, 2011, pp. 81663O-1-81663O-12.
- **Tuck-Boon Chan**, John Sartori, Puneet Gupta and Rakesh Kumar, "On the Efficacy of NBTI Mitigation Techniques", *Proc. Design, Automation and Test in Europe*, 2011, pp. 1-6.
- **Tuck-Boon Chan**, Aashish Pant, Lerong Cheng and Puneet Gupta, "Design-Dependent Process Monitoring for Back-End Manufacturing Cost Reduction", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2010, pp. 116-122.
- **Tuck-Boon Chan**, Abde Ali Kagalwalla and Puneet Gupta, "Measurement and Optimization of Electrical Process Window", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2010, pp. 76410J-1-76410J-11.
- **Tuck-Boon Chan** and Puneet Gupta, "On Electrical Modeling of Imperfect Diffusion Patterning", *Proc. IEEE/ACM International Conference on VLSI Design*, 2010, pp. 224-229.

ABSTRACT OF THE DISSERTATION

Mitigation of Variability and Reliability Margins in IC Implementation

by

Tuck Boon Chan

Doctor of Philosophy in Electrical Engineering (Computer Engineering)

University of California, San Diego, 2014

Professor Andrew B. Kahng, Chair

In the late-CMOS era, system-on-chip design and manufacturing margins continue to increase in light of process variability, circuit reliability and wide operating conditions. Despite continuing enhancements to both manufacturing and design technologies, substantial IC product value in terms of manufacturing yield, circuit area and power, and design turnaround time is left on the table due to conservatism in the design and manufacturing flows. These margins are now extremely costly, as the benefits from deployment of the next technology node are now only approximately 20% in circuit performance, power and density. To reduce margins, accurate modeling and assessment of the impacts of variability and reliability are essential. Meanwhile, innovative manufacturing and design techniques must be developed based on a comprehensive understanding of the benefits and costs of such new measures. This thesis presents new techniques to mitigate variability and reliability margins in leading-edge SoC design and manufac-

turing. These techniques can be grouped into three main thrusts: (i) design for manufacturability and reliability; (ii) signoff condition optimization; and (iii) design-aware manufacturing optimization.

In the *design for manufacturability and variability* thrust, this thesis presents two performance sensor designs for adaptive voltage scaling, which can be used to mitigate the impact of process variations. To reduce design margins for time-dependent dielectric breakdown reliability, this thesis presents a layout optimization technique and a design-dependent reliability analysis framework.

In the *signoff condition optimization* thrust, this thesis presents analyses of the design overheads due to suboptimal signoff conditions with respect to (i) circuit operating voltage and performance; (ii) modeling of timing impacts of circuit aging; and (iii) corner models of wire parasitic resistance and capacitance. Tradeoffs between design quality and signoff margins, as well as methods to optimize signoff conditions, are also addressed.

In the *design-aware manufacturing optimization* thrust, this thesis presents three distinct techniques to improve manufacturing yield by considering the impact of manufacturing variations on the design's timing and leakage power. First, the *electrical process window* provides a more accurate method to quantify the impact of lithographic variability on circuit performance and leakage. Second, *design-dependent monitoring* provides a cost-effective way to estimate circuit parametric yield based on test structures deployable in the early stages of a manufacturing flow. Finally, analysis of the impact of overlay error in double-patterning lithography provides guidelines to reduce circuit performance variation.

Chapter 1

Introduction

The maximum on-chip clock frequency of *microprocessor* (MPU) and *system-on-chip* (SoC) IC product classes has been a key metric of semiconductor technology scaling. Figure 1.1 shows how the frequency roadmap of the *International Technology Roadmap for Semiconductors* (ITRS) [233] has been slowing down [103], closely reflecting product data from the Stanford CPUDB repository [246]. The evolution of the MPU frequency roadmap may be deconstructed as follows. Before 2001, aggressive architectural pipelining and device improvements double clock frequency per technology node (41%/year improvement). Starting in the early 2000s, from the point where the pipelining knob runs out of steam (only $\sim$12 fanout-of-4 inverter delays can practically fit in a clock cycle), frequency scaling is based solely on device speed improvement, and reduces to 17%/year. The frequency scaling is subsequently constrained to 8%/year as products reach the power limits of the high-performance MPU (desktop or server) platform, and is further slowed as transistor performance improvement comes at too high a cost in leakage power at the most recent technology nodes.

Designers continue to extract value from Moore's Law by scaling density (i.e., layout area per DRAM bit, SRAM bitcell, or logic gate) even when frequency scaling has slowed down. Density scaling is mainly driven by lithography improvements, which reduce the minimum *metal pitch* (i.e., a wire width plus a wire spacing). When the metal pitch scales by 0.7$\times$ in both the horizontal and vertical dimensions of a two-dimensional layout, the area scales by 0.49$\times$, and the "available" density is approximately doubled. In the past decades, such 0.7$\times$ *geometric scaling* in each successive technology node [234] has enabled doubling of transistor count in a constant die area. However, the data in Figure 1.2 shows that although lithography has delivered the "available" 2$\times$ per node density scaling, the "actual" density scaling in products has slowed

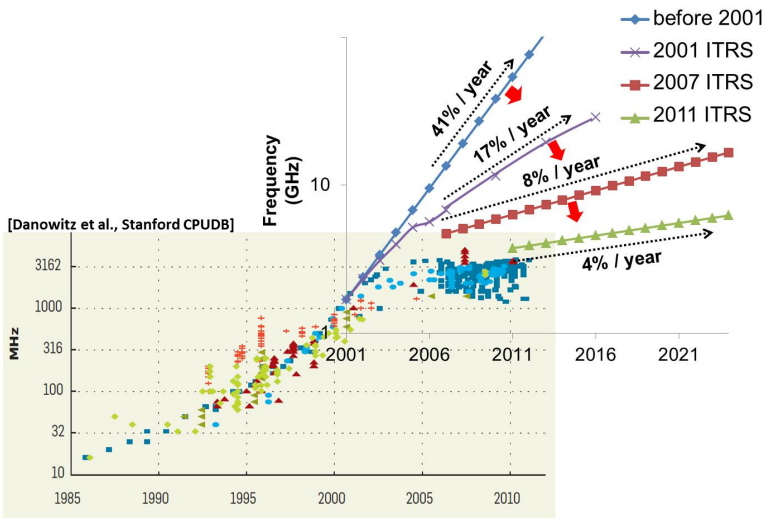


Figure 1.1: Overlay of the ITRS [233] maximum on-chip clock frequency roadmap with data from the Stanford CPUDB repository [246]. This figure is reproduced from [103].

down to 1.6× per node since 2007. Such frequency and density scaling trends (i.e., Figures 1.1 and 1.2) are indicators of a *late CMOS era* in which the recent benefits of technology scaling are significantly less than what would have been expected according to historical trends. Today, moving to a new technology node is very costly, and returns on investment are unclear as it becomes more challenging to obtain even 20% improvements in power, performance and area at the new node. In this regime, SoC product companies cannot afford to overlook or sacrifice even a small percentage of potentially available power, performance or area improvements. This motivates the focus of this thesis research on mitigation of *margins* in IC implementaion.

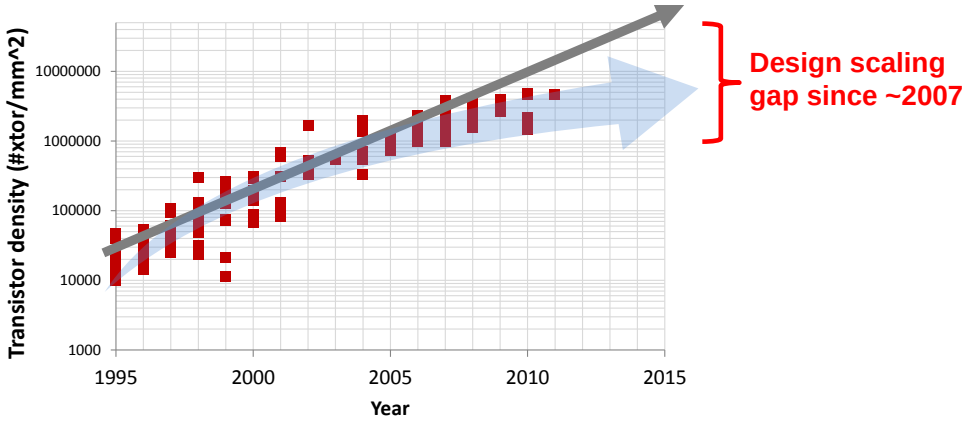


Figure 1.2: Gap between "available" density scaling (gray arrow) and "actual" density scaling (red squares) [103].

In the remainder of this chapter, we review background of manufacturing variability and reliability in Sections 1.1 and 1.2 respectively. Section 1.3 discusses issues related to over-margining. Section 1.4 discusses previous techniques in the three areas of manufacturing-aware design, design-aware manufacturing, and adaptivity mechanisms. Section 1.5 presents the organization of the remainder of this thesis.

1.1 Variability

To mitigate the impacts of process, voltage and temperature variations on circuit performance, a circuit is typically overdesigned to ensure that it will function correctly across all possible maanufacturing and operating contexts. This incurs circuit area and power overheads. In this section, we review sources of variations and the corresponding variability models.

1.1.1 Process Variation

Process variation has been a critical aspect of semiconductor manufacturing [117]. When new process technologies are introduced, process variation causes manufactured chips to exhibit wide performance spread [21], with yields of good die as low as 30% to 50% [207]. The process variations can be broadly classified into those that occur in the *front-end-of-line* (FEOL) and those that occur in the *back-end-of-line* (BEOL).

In the FEOL, variations in transistor gate length, gate oxide thickness, transistor-channel doping, etc. cause transistor delay and leakage variations. To model the FEOL variations in a digital IC implementation flow, several different methods may be used. The most widely used, conventional method provides a set of *corners* with biased process parameters to represent the impact of process variations on transistor delay and leakage. Since variations in the NMOS and PMOS transistors may be different, the foundry usually provides combinations of slow (S) and fast (F) corners, as well as a typical (T) corner: $\{$SS, SF, TT, FS, FF$\}$. The FEOL corners are depicted in Figure 1.3. Note that each of these FEOL corners corresponds to an ordered pair of (NMOS, PMOS) device models. The key aspect of this corner-based approach is the implied assumption that the FEOL variations are bounded within the "dotted box" defined by the four extreme corners as shown in Figure 1.3.

As the number of independent and significant variation sources increases, design signoff using a corner-based *static timing analysis* (STA) may become both pessimistic (i.e., leading to overdesign that wastes area and power for a given level of performance) and risky (i.e., with

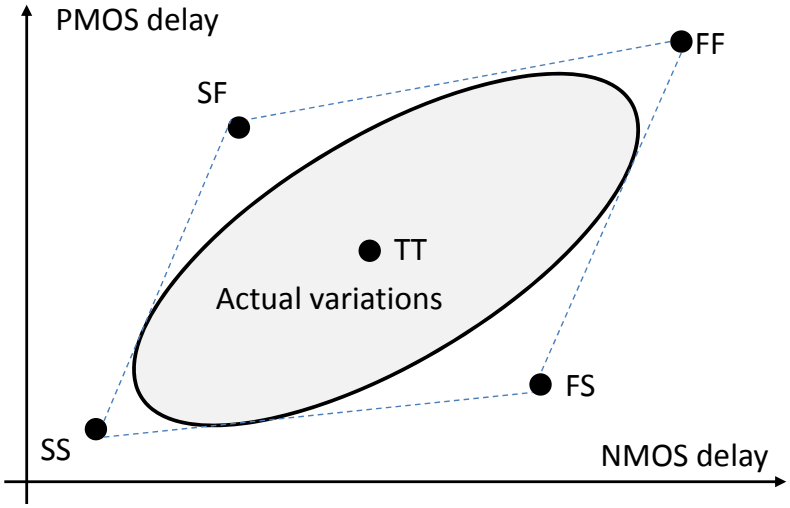


Figure 1.3: Illustration of typical FEOL corners.

some manufactured die still failing to meet timing requirements) [201]. This challenge can be addressed by using *statistical static timing analysis* (SSTA) in which the gate delays are specified as distributions rather than as deterministic values. By calculating the distribution of signal arrival times at timing endpoints, timing slack distributions can be computed with a prescribed confidence level. Initial works on SSTA date from the 1960s [114] and the early 1990s [14] [22] [62] [98]. Subsequently, many algorithmic innovations have improved the efficiency and accuracy of SSTA [2] [5] [43] [78] [163] [202]. Although SSTA can provide more accurate timing analysis, it has not been widely adopted by industry due to such technical issues as interconnect analysis, coupling noise, and complex delay modeling [18]. More importantly, fabless design houses and silicon foundries lack a business model that permits accurate communication of the statistics of the manufacturing process [18].

An alternative method to account for timing variability intentionally multiplies the delay of a data path and/or the clock latency by a *derating factor*, so as to finely tune the timing margins in the design. In an *advanced on-chip variation* (AOCV) methodology [115] [153] [249], timing analysis tools can extract the topology of a netlist and assign stage- and/or location- dependent derating factors accordingly. This implicitly allows designers to apply statistical timing methods to reduce the pessimism of the corner-based approach. For example, designers may apply a stage-dependent derating factor which decreases as the number of stages in a data path increases, i.e., the delay variation on a long data path becomes smaller due to the averaging of random variations.

The back-end-of-line in the IC manufacturing process fabricates a stack of metal and dielectric layers. Figure 1.4 shows a cross-section with three metal layers (M1, M2 and M3). W, T and H are the metal width, metal thickness and dielectric thickness, respectively. The parasitic resistance (R) and capacitance (C) variations in BEOL are typically modeled by BEOL corners in which all BEOL layers vary in the same way [91]. For example, Table 1.1 shows common BEOL corners (Y) in which the wire width (ΔW), wire thickness (ΔT) and dielectric thickness (ΔH) variations are biased to the minimum or maximum values to capture the extreme conditions. Such BEOL corners are very pessimistic because the probability that all BEOL layers are simultaneously skewed towards the extreme condition is extremely small. As wire geometries continue to shrink with each new process node, RC variations in BEOL have become major sources of variation especially when the gate delay is small (e.g., at high operating voltage) [155].

Table 1.1: Typical BEOL corners with skewed parameters.

Corner	ΔW_m	ΔT_m	ΔH_m
Y_{typ}	typical	typical	typical
Y_{cb}	minimum	minimum	maximum
Y_{cw}	maximum	maximum	minimum
Y_{rcb}	maximum	maximum	maximum
Y_{rcw}	minimum	minimum	minimum

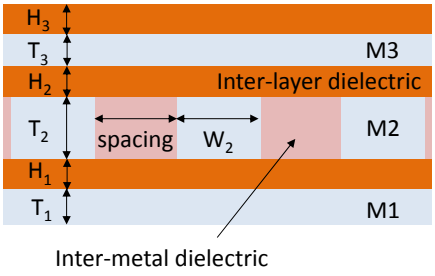


Figure 1.4: Illustration of the cross-section of a typical metal stack.

Both FEOL and BEOL variations can have *systematic* and *random* components. *Systematic variation* is the portion of the variation which can be deterministically modeled. Other variations which cannot be modeled are lumped and denoted by *random variation* [49].

1.1.2 Lithography

Lithography in IC manufacturing is a major source of both FEOL and BEOL variations. The rapid pace of semiconductor scaling over the last decades, coupled with much slower advances in lithography technology, has forced 193nm optical lithography beyond its limit. Since the availability of the *extreme ultraviolet lithography* (EUVL) remains unclear, double-patterning lithography (DPL) [67] has been adopted at the 20nm logic half-node (sub-80nm pitch), and it remains a strong candidate for BEOL patterning in more advanced technology nodes. There are generally two kinds of DPL, *litho-etch-litho-etch* (LELE) and *spacer-assisted double patterning* (SADP) [67] [187].

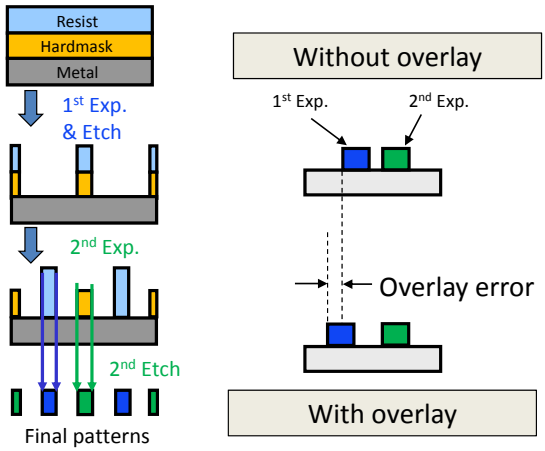


Figure 1.5: Overlay in LELE double patterning

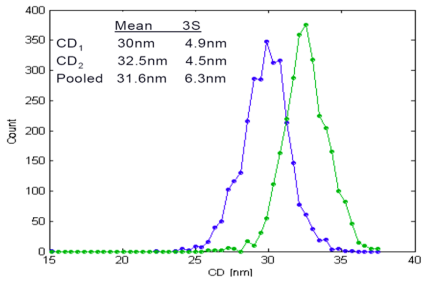


Figure 1.6: Bimodal CD variation in LELE. The figure is reproduced from [67].

In LELE double-patterning lithography, layout features with a spacing less than the *minimum coloring spacing* are assigned opposite colors. As illustrated in Figure 1.5, a set of layout features with the same color are patterned through a lithography step (i.e., exposure and etch) followed by a similar lithography step for the second set of layout features. Because of the two

separate lithography steps, the pitch of the features in a given mask layout is effectively "doubled" i.e., significantly relaxed, with respect to the pitch of features in the final fabricated design. However, overlay error can lead to two distinct distributions of *critical dimensions* (CD) depicted in Figure 1.6. Such a bimodal CD distribution [35] [92] [102] causes large CD variations that induce wire width and spacing variations in the BEOL, and gate length variations in the FEOL.

In SADP, layout features are decomposed into the mandrel and trim masks. As illustrated in Figure 1.7, each mandrel pattern will be surrounded by spacer material. Note that additional mandrel features are added to control the spacing between spacers. As shown in the figure, the final pattern will be formed by the region which is covered by the trim mask, but not covered by spacer Compared to LELE, SADP has relatively smaller overlay, but the mask decomposition problem is more complex [66].

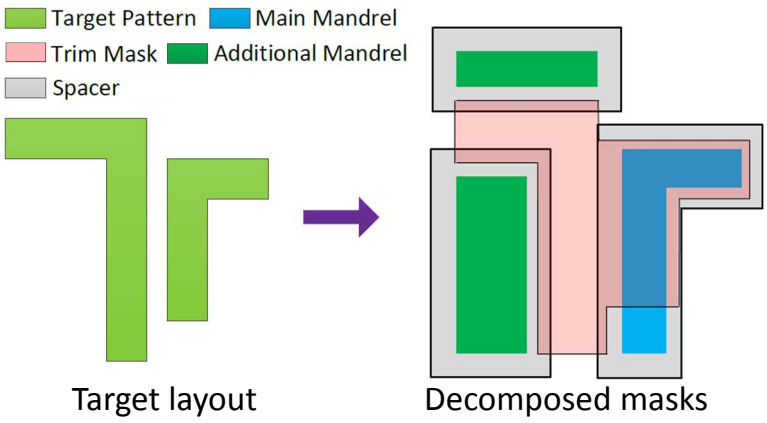


Figure 1.7: Example of SADP decomposition. This figure is reproduced from [66].

On the other hand, lithographic *resolution enhancement techniques* (RETs) such as *optical proximity correction* (OPC), subresolution assist features and phase shift masks have become a necessity to ensure the printability of subwavelength feature sizes [113] [134] [133]. Despite the RETs, there are CD variations because of varying exposure, focus or overlay in lithography. To account for and, more importantly, to bound these CD variations, the *process window* represents a (Cartesian product of) range(s) of manufacturing process parameters, such that chips produced by a process that remains within these tolerances will meet desired specifications [141]. Typically, the process window is defined so as to ensure that the CD of any feature dimension does not deviate from its nominal value by more than a predefined (dimensional, geometric) tolerance [133] [141].

1.1.3 Environmental Variations

Supply voltage (V_{dd}) and temperature variations are the common environmental variations in IC. Depending on the circuit activities at different parts of a chip, there can be nonuniform current demand across the chip. V_{dd} variation occurs when the nonuniform current demand, in conjunction with the design of the power distribution network, causes a nonuniform *IR drop* (IR drop is the V_{dd} difference due to the product of current demand and resistance in the power distribution network). For example, Figure 1.8 shows that the ΔV_{dd} varies across the chip. Meanwhile, because of the power dissipated by circuit activity, there is also temperature variation across the chip. Such V_{dd} variation can be mitigated through synthesizing a more robust power delivery network at the cost of routing resources.

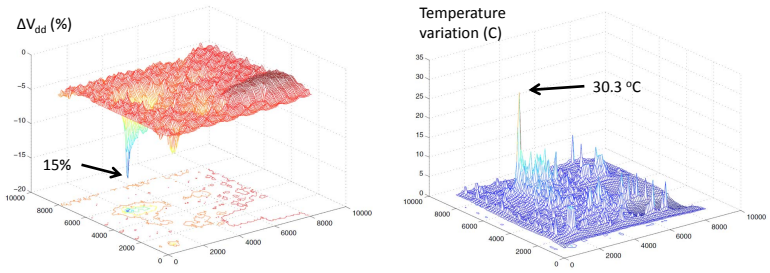


Figure 1.8: Voltage and temperature variations for a chip. This figure is reproduced from [63].

1.2 Reliability

A reliable IC must be able to withstand various *wearout mechanisms* throughout its lifetime. The major wearout mechanisms in advanced CMOS technology include *bias temperature instability* (BTI), *gate oxide breakdown*, BEOL *time-dependent dielectric breakdown* (TDDB), *electromigration* (EM), *hot carrier injection* (HCI) and *stress-induced voiding* (SIV). BTI, HCI and gate oxide breakdown are FEOL reliability issues which affect transistors' performance or operation. TDDB, EM and SIV are BEOL reliability issues which can cause shorts or opens in interconnects. Among these wearout mechanisms, we focus on FEOL BTI and BEOL TDDB because they are well-recognized as critical obstacles for technology scaling [13] [206].

1.2.1 Bias Temperature Instability

BTI is manifested as an increase in a transistor's threshold voltage ($|V_{th}|$) and, consequently, as an increase in transistor delay, whenever a transistor is under stress, i.e., when the

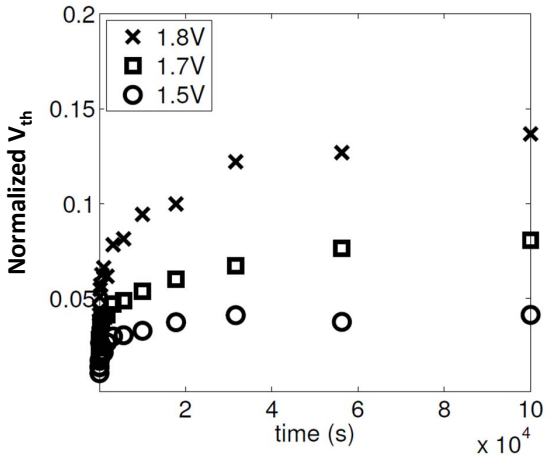


Figure 1.9: The rate of NBTI degradation is fast initially and slows down considerably under continued stress. This figure is reproduced from [203].

voltage at a transistor's gate ($|V_{gs}|$) is larger than $|V_{th}|$. Relaxation of the stress ($V_{gs} = 0$) can recover only part of the V_{th} degradation [3], causing an overall increase in transistor delay over time (i.e., BTI degradation). If not appropriately provisioned for, increased delay can result in timing failures on critical paths [71]. The BTI effect can occur in both NMOS and PMOS transistors; respectively, this is known as *positive-bias temperature instability* (PBTI) and *negative-bias temperature instability* (NBTI). BTI degradation is frequency-independent but increases with electric field and temperature [3] [197]. Also, due to the underlying physical phenomena that cause BTI, the degradation is "front-loaded" by nature. As illustrated in Figure 1.9, the rate of NBTI degradation is rapid during the early lifetime of the transistor and slows down considerably under continued stress. As technology scales, the increased electric field across the gate oxide [128] [235] accelerates BTI degradation [76].

1.2.2 Back-End-of-Line Time-Dependent Dielectric Breakdown

In an integrated circuit, signals are transmitted using interconnects (i.e., vias and wires) fabricated on metal layers. The interconnects in an IC are isolated by insulating dielectric material. The insulating property of a dielectric degrades when there is an electric field across the dielectric. Dielectric breakdown occurs when the degraded dielectric eventually forms a conducting path between interconnects. Such time-dependent dielectric breakdown causes a short circuit between interconnects and consequently results in a malfunctioning IC. Dielectric *mean time to failure* (MTTF) is defined as the expected time for the dielectric to form a conductive path between interconnects. The MTTF decreases as the electrical field across the dielectric

increases [15] [44] [56] [160] [183]. In other words, the MTTF of dielectric between interconnects decreases when the voltage difference between the interconnects increases or when the spacing between the interconnects decreases. Thus, TDDB will be a major reliability concern for BEOL dielectric due to the increasing electric field as technology scales. Indeed, at the $20nm$ node (sub-$70nm$ local metal pitch) with LELE double patterning, TDDB reliability is a primary limiter to further wiring density improvement [13].

1.3 Problem: Over-Margining

Within the IC design process, *signoff* is a fundamental part of the design closure stage in which designers perform a set of canonical analyses based on models provided by the silicon foundry. If the design passes the analyses, then the assumption is that the manufactured chip will meet all functional and performance specifications. The analyses span a range of design criteria, including functionality, timing, power, reliability, etc. To account for circuit variability and reliability, designers tend to insert margins in the signoff analyses based on worst-case scenarios. Although it is necessary to have margins to cover uncertainties in the design and manufacturing steps, excessive margins reduce the benefits realized from technology scaling. Figure 1.10 illustrates the scenario in which the nominal design quality improves with each successive technology node. However, the signoff margin also increases due to increasing variability and reliability constraints, as well as pessimism in the circuit design implementation. As a result, the design can only gain a small fraction of the benefits from technology scaling. Since available, potential benefits from a given next-generation technology are already small in the late-CMOS era, the lost benefits due to over-margining are costly. For example, Weckx et al. [206] show that because of the increasing V_{th} margin (due to both process variation and BTI degradation), it may be not beneficial for certain products to move to an advanced technology.

1.4 Previous Design and Manufacturing Optimization Techniques

We now review three relevant classes of techniques that mitigate the impacts of variability and reliability: (i) *manufacturing-aware design* (MAD) techniques which focus on compensating process variation; (ii) *design-aware manufacturing* (DAM) techniques which exploit design-side information to improve manufacturing yield; and (iii) *adaptive body biasing* (ABB) or *adaptive voltage scaling* (AVS) techniques which adaptively compensate for circuit performance variation.

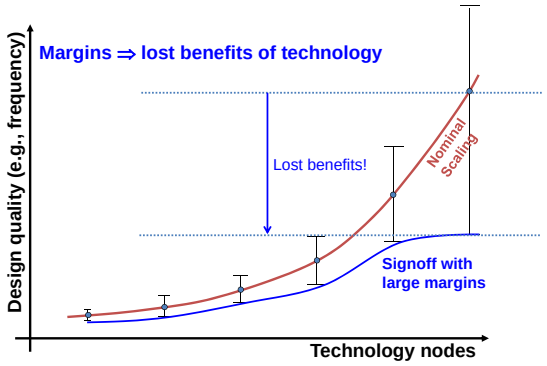


Figure 1.10: Excessive margin reduces the benefits of technology scaling.

1.4.1 Manufacturing-Aware Design

MAD is a paradigm in which design techniques are optimized based on better understanding of manufacturing processes such as lithography, etching, chemical-mechanical polishing, etc. Since the $90nm$ technology node, many new MAD methods have been conceived and deployed to mitigate increasing lithography-induced variation, as well as other variability issues that are not practically addressable through manufacturing process control [85]. Following are some examples of MAD techniques.

Focus Variation-Aware Design Techniques

Focus variation is one of the major sources in lithography which can occur due to changes in wafer flatness or lens imperfections. When there is focus variation, linewidth decreases for isolated lines (i.e., lines with large spacings) but increases for dense lines, as shown in Figure 1.11. Such linewidth variation can lead to transistor gate-length variation, which affects circuit performance. To mitigate the focus-dependent gate-length variation, Gupta et al. [85] propose a MAD methodology which mixes isolated and dense lines either across transistors within a cell or across cells on critical paths. Because of the cancellation of gate-length variations within cells or along critical paths, a circuit implemented using this MAD technique is insensitive to focus variation. This example shows that through a better understanding of the lithography process, it is possible to improve the design methodology to mitigate the impact of process variability. Note that the MAD concept can be applied to other circuit properties. For example, Kahng et al. [106] propose a leakage optimization method based on the same focus-dependent linewidth variation model.

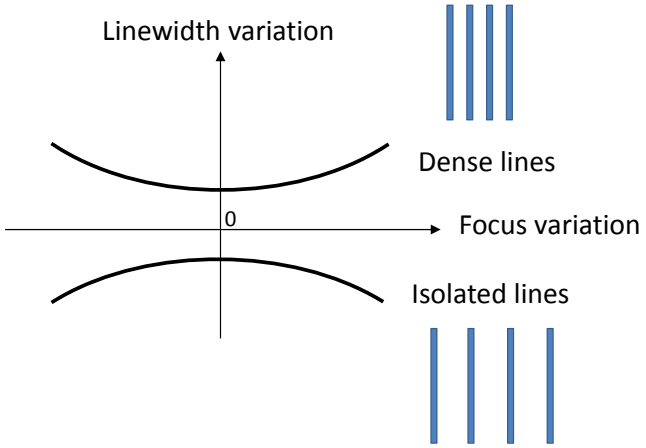


Figure 1.11: Idealized Bossung plot representing linewidth variation with respect to focus variation for isolated and dense lines.

Shallow-Trench Isolation Aware Design Techniques

Shallow-trench isolation (STI) is a commonly used technology that isolates the active regions of CMOS transistors. Because of STI-induced stress, the mobility of a transistor varies depending on the distance between the transistor channel and the nearest boundary of an STI region. In a digital design, this distance is determined by the placement of the standard cells. By modeling the impact of STI-induced stress, Kahng et al. [108] propose a STI width-aware design methodology which perturbs cell placement and inserts dummy features on the active layer to optimize circuit timing. A complementary work of Joshi et al. [97] proposes a stress-aware standard cell layout optimization that enhances the drive currents of transistors with minimal cell area penalty.

Layout Regularity

Due to imperfections in lithography and other process steps, the geometric dimensions of a fabricated transistor can deviate from the drawn transistor shape and affect circuit performance (e.g., due to STI-induced stress, well proximity, or short-channel effects). Such lithography-induced variation can be (partially) compensated when the design layout is restricted to regular patterns through strict design rules. For example, Figure 1.12 shows a classical flip-flop layout versus a regular layout for the same flip-flop. Although the regular layout may consume more silicon area, the layout regularity helps to reduce process variations because manufacturing processes can be heavily optimized for a smaller set of layout pitches and patterns. The reduced

process variations translate to smaller margins in design implementation, which compensate for the increased silicon area in the regular layout [88]. A regular layout methodology can also improve manufacturing yield as it helps to avoid layout patterns that are prone to form short or open circuits.

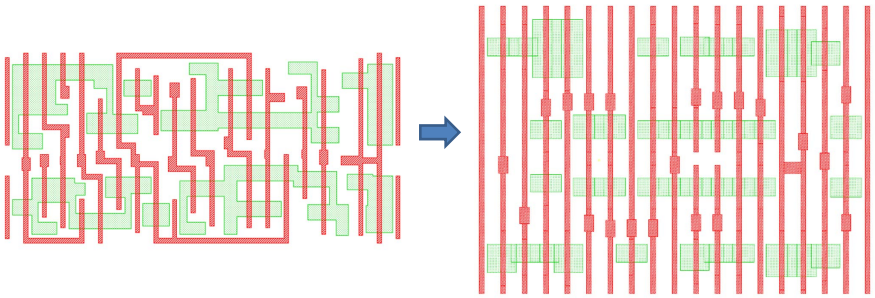


Figure 1.12: A classical flip-flop layout (left) versus a regular layout. This figure is reproduced from [88].

1.4.2 Design-Aware Manufacturing

With subwavelength lithography, transistor channels become nonrectangular, which causes circuit performance and power variations [33]. To address this, many so-called electrically-driven techniques have been proposed. Zhang et al. [216] develop an analytical model to account for corner rounding in printed transistors and use the model to drive the OPC step. A similar method is also described in [9]. Further, Gupta et al. [86] use timing slacks obtained from the critical paths to reduce the complexity of post-OPC mask shapes. These methods achieve smaller circuit-performance variation and reduced mask complexity despite large geometric errors [176].

Another type of approach is exemplified by the work of Jeong et al. [94], who propose to exploit the capability of *exposure dose control* in the lithography system to optimize timing yield and leakage power. Given the placement of standard cells in a design and a corresponding timing analysis, the lithography exposure field is partitioned into a set of grids in which the exposure dose is optimized based on the timing criticality of the cells in the corresponding grid. For example, if a grid has standard cells on a *setup timing-critical* path, the grid will be given a larger exposure dose. As a result, gate lengths of the transistors in the grid will be smaller, and the cell delays are reduced. Similarly, the leakage power of the transistors can be reduced by selectively applying a smaller exposure dose in the grids without setup timing-critical paths.

Since the size of mask defects shrinks along with smaller layout feature size, mask

inspection tools must use smaller pixels and/or stronger sensitivities in order to detect mask defects, increasing runtime and manufacturing costs. To address this issue, Kagalwalla et al. [101] propose a design-aware mask inspection method to locate nonfunctional features in a post-OPC layout. Based on this design-derived information, the mask inspection tool can use the appropriate pixel size and sensitivity to reduce false and nuisance defects without missing any critical defects. As a result, the mask inspection time and writing cost can be reduced.

1.4.3 Adaptive Design Techniques

As technology scales, the impact of random variation increases [182]. Since the random variations are unpredictable (e.g., process variations vary for different chips), the impact of random variations cannot be compensated easily by MAD techniques. Furthermore, variation of operating conditions such as ambient temperature, or aging phenomena such as BTI, will affect performance and power consumption throughout chip lifeimte. To address these challenges, adaptive circuits have been proposed.

Adaptive Body Biasing

The bodies of PMOS and NMOS devices in a digital circuit are normally connected to the supply (V_{dd}) and ground (V_{ss}) voltages, respectively. When the body voltages are biased to different values, the *body voltage biasing* effect changes the V_{th} of the transistors. This phenomenon provides a useful knob with which to adjust the V_{th} of a circuit. A circuit with adaptive body biasing (ABB) is designed such that the body voltages of the circuit can be properly calibrated post-silicon fabrication or dynamically adjusted during runtime [127]. Since the body bias of each die can be adjusted independently, even a die with strongly skewed V_{th} due to manufacturing variation can be adjusted to meet the power and performance specifications. Such ABB-enabled V_{th} adjustment techniques can be applied to optimize circuit performance and power consumption [122] [144] and improve manufacturing yields [195] [118]. Zhuo et al. [221] show that the benefits of a circuit with post-silicon ABB can be further improved through gate sizing and optimization of cell clustering for fine-grained ABB.

In advanced technology nodes, applying ABB becomes more difficult because the effect of body biasing becomes less significant. However, it is possible to increase the effect of body biasing through improved device engineering [159] [164].

Adaptive Voltage Scaling

AVS is a design methodology that regulates the supply voltage of an IC to optimize circuit performance and power consumption, as well as compensate for circuit wearout [11] [34] [65] [69] [119] [217]. Figure 1.13 shows different AVS approaches. The simplest AVS implementation is to scale V_{dd} based on a precharacterized lookup table (LUT) [144]. This implementation leaves a large design margin because the LUT implicitly guardbands for process and temperature variations. To reduce margin due to process variation, post-silicon characterization can be used [196]. However, this kind of AVS can only compensate for process variation.

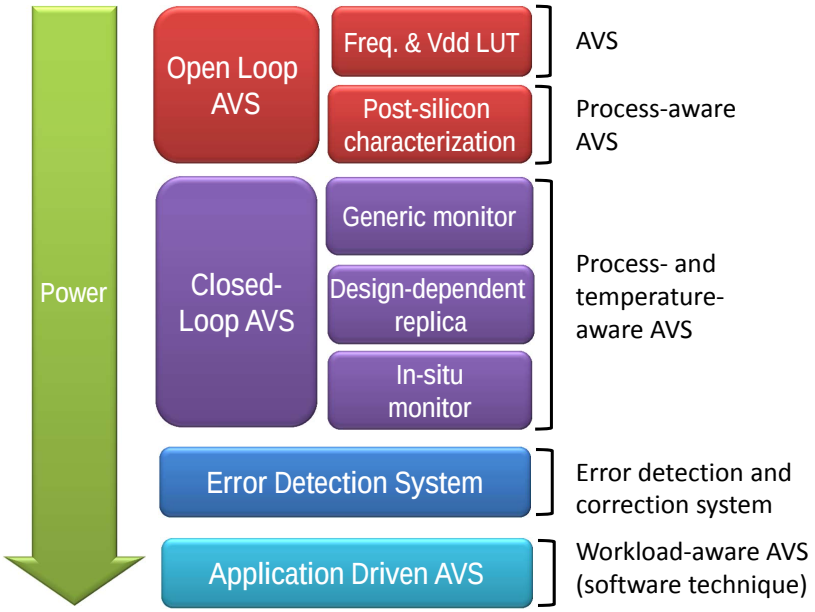


Figure 1.13: Different classes of adaptive voltage scaling techniques.

Closed-loop AVS is another class of AVS technique which has on-chip monitors to capture both process and temperature variations. In a closed-loop AVS, the monitor design is the key to success. The monitors can range from a generic *ring oscillator* (RO) [27] [36] or design-specific monitor [34] [65] [69] to an in-situ monitor which measures actual path delays [73]. Alternatively, one may also design the monitoring circuit to have error detection and correction abilities, enabling scaling of V_{dd} until an error occurs, so as to minimize the design margin [59] [194]. Beyond hardware techniques, AVS can also optimize the frequency and V_{dd} simultaneously based on instructions from software or the user [135].

When a chip runs faster than the required performance (e.g., due to process variation), AVS can be used to reduce the dynamic power. However, using AVS alone may be limited in

reducing the leakage power. Since ABB is more effective in managing leakage power consumption, AVS and ABB can be used simultaneously to control both dynamic and leakage power [144] [170].

1.5 This Thesis

To mitigate the margins for variability and reliability, innovative design and manufacturing techniques are urgently required by the semiconductor inductry, and have been developed in the course of this thesis researach. Figure 1.14 illustrates the scope and organization of this thesis, showing the grouping of variability and reliability mitigation techniques into three main thrusts which respectively correspond to the following three chapters:

- Design for manufacturability and reliability;
- Signoff condition optimization;
- Design-aware manufacturing optimization.

In the *design for manufacturability and reliability* thrust, this thesis presents two *performance monitors* for adaptive voltage scaling to mitigate process variation. To mitigate the margins for time-dependent dielectric breakdown, improved design-dependent reliability analysis and layout optimization techniques are included.

In the *signoff condition optimization* thrust, this thesis presents analyses on the design overheads due to suboptimal signoff conditions in (i) circuit operating voltage and performance, (ii) circuit aging timing models, and (iii) wire resistance and capacitance models. Meanwhile, the tradeoffs between design quality, signoff margins, and methods to optimize signoff conditions are also included.

In the *design-aware manufacturing optimization* thrust, this thesis presents three distinct techniques to improve manufacturing yield by considering the impact of manufacturing variations on the design's timing and leakage power. First, the concept of *electrical process window* provides a more accurate method to quantify the impact of lithographic variability on circuit performance and leakage. Second, *design-dependent monitoring* provides a cost-effective way to estimate circuit parametric yield based on test structures available in the early stages of a manufacturing flow. Finally, analysis of the impact of overlay error in double-patterning lithography provides guidelines to reduce circuit performance variability.

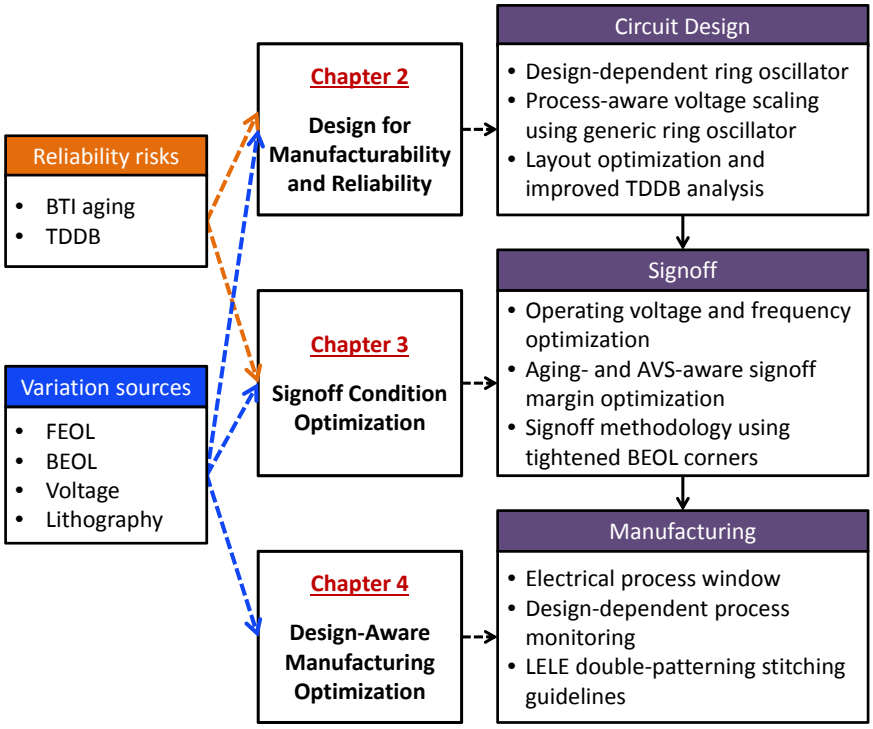


Figure 1.14: Scope and organization of this thesis.

The remainder of this thesis is organized as follows.

- Chapter 2 presents three distinct techniques for the mitigation of variability and reliability margins. First, we propose a systematic method to synthesize *design-dependent ring oscillators* (DDROs) which track design-specific performance variation. The DDROs can be used for process monitoring and as performance monitors for *adaptive voltage scaling* (AVS). To design the DDROs, we extract the sensitivities of critical path delay to process variation sources. Based on the extracted sensitivities, we synthesize the DDROs using an integer linear program (ILP). To validate the DDRO concept, we implement a testchip using 45nm silicon on insulator (SOI) technology. Second, we propose an alternative RO design for *process-aware voltage scaling* (PVS). Instead of designing performance monitors to track the timing performance of critical paths, we design the PVS ROs such that they require a relatively higher supply voltage compared to critical paths of an SoC so as to compensate process variation-induced circuit performance drift. Therefore, any SoC manufactured in the process can safely perform a closed-loop AVS by using these ROs as hardware performance monitors. Third, we propose to improve dielectric relia-

bility through a post-layout optimization. In the layout optimization, we locally shave and/or shift a fraction of wire width to increase the spacing between wires and/or between adjacent-layer vias and wires. Separately, we propose a signal-aware chip-level TDDB reliability estimation method which provides less pessimistic estimates of TDDB risk.

- Chapter 3 presents various techniques to optimize aspects of signoff, including (i) *operating mode* (i.e., an (operating frequency, voltage) pair), (ii) aging margin, and (iii) back-end-of-line (BEOL) corners. First, we propose a concept of *mode dominance* (see Section 3.1 for the detailed definition) which can be used as a guideline for signoff mode selection. Further, we propose a scalable, model-based adaptive search methodology for signoff mode selection. Second, to optimize the aging margin for a circuit with adaptive voltage scaling (AVS), we study the conditions under which a circuit with AVS requires additional timing margin during signoff. Then, we propose two heuristics for chip designers to characterize an aging-derated standard-cell timing library that accounts for the impact of AVS during signoff. Further, we compare circuits implemented with the aging-aware signoff method based on aging-derated libraries against those based on a *flat timing margin*. Third, to reduce timing margin for BEOL variations, we first analyze the pessimism in the conventional BEOL corner. From observations of the circuit properties of timing-critical paths, we propose a method to identify the paths which can be safely signed off using tightened BEOL corners that embody reduced pessimism.

- Chapter 4 presents three distinct techniques for manufacturing optimization. First, we introduce a method to calculate the *electrical process window* (EPW) of a design which accounts for electrical specifications. The EPW is more accurate and less pessimistic compared to the conventional *geometric process window*, which only considers CD variation. We analyze various layout-transparent methods to enlarge the EPW to improve manufacturing yield. We also propose approximate methods to evaluate the EPW; these can be used with little or no design information. Furthermore, we propose a method to extract *representative layouts* for large designs which can then be used to evaluate the EPW with much smaller runtime. Second, we propose a design-dependent process monitoring strategy which can predict design performance based on measurements obtained from test structures in wafer scribelines. Since these measurements are available in the early stages of manufacturing, we propose to use the predicted design performance to prune bad wafers. Such early pruning can save test and back-end manufacturing costs. Third, we study the impact on BEOL electrical performance of stitching locations in LELE double-patterning

mask design. We derive analytical RC equations to model the impact of CD variation due to the overlay error in LELE double patterning. Based on the analytical equations, we propose guidelines for optimal stitching to reduce RC variations.

Chapter 2

Design for Manufacturability and Reliability

This chapter presents three distinct techniques for the mitigation of variability and reliability margins. First, we propose a systematic method to synthesize *design-dependent ring oscillators* (DDROs) which track design-specific performance variation. The DDROs can be used for process monitoring and as performance monitors for *adaptive voltage scaling* (AVS). To design the DDROs, we extract the sensitivities of critical path delay to process variation sources. Based on the extracted sensitivities, we synthesize the DDROs using an integer linear program (ILP). To validate the DDRO concept, we implement a testchip using $45nm$ silicon on insulator (SOI) technology. Second, we propose an alternative RO design for *process-aware voltage scaling* (PVS). Instead of designing performance monitors to track the timing performance of critical paths, we design the PVS ROs such that they require a relatively higher supply voltage compared to critical paths of an SoC so as to compensate process variation-induced circuit performance drift. Therefore, any SoC manufactured in the process can safely perform a closed-loop AVS by using these ROs as hardware performance monitors. Third, we propose to improve dielectric reliability through a post-layout optimization. In the layout optimization, we locally shave and/or shift a fraction of wire width to increase the spacing between wires and/or between adjacent-layer vias and wires. Separately, we propose a signal-aware chip-level TDDB reliability estimation method which provides less pessimistic estimates of TDDB risk.

2.1 Design-Dependent Ring Oscillator (DDRO) Performance Monitors

Circuit performance variability continues to increase due to process variation, wide operating ranges, and other factors. Performance variability can often be compensated by accurate circuit performance estimation and subsequent adaptation. For example, (i) circuit performance can be monitored in the manufacturing flow for process tuning, or (ii) systems with adaptive mechanisms can optimize the tradeoff between energy and performance based on feedback from runtime circuit performance monitors [83]. We define circuit *performance monitoring* as a process which estimates the worst-case delay of a circuit, based on the measurements obtained from on-chip monitors.

Generic monitors range from simple inverter-based *ring oscillators* (ROs) to more sophisticated process-sensitive ROs (PSROs) [17] [142] and alternative monitoring structures such as phase-locked loops (PLLs) [109]. However, such generic monitors are inadequate for capturing design characteristics such as the mix of device types, which differ in responses to process variations, on critical paths. As a result, delay estimation using generic monitors is less accurate, which leads to larger timing margins.

Design of monitoring structures that are correlated to circuit performance (*design-dependent monitors*) has been addressed in several ways. Liu and Sapatnekar [137] propose a method to synthesize a single *representative critical path* (RCP) for post-silicon delay prediction. The RCP is designed such that it is highly correlated to all the critical paths for some expected process variations. This approach uses only a single RCP to estimate the worst-case delay of multiple critical paths. Since the critical paths may have different sensitivities to process variations, using a single RCP may be inaccurate. The *tunable replica circuit* (TRC) method in [65] synthesizes different delay paths to more flexibly mimic circuit performance, but has larger design overhead compared to RO approaches. TRC also requires costly calibration to obtain configurations that correspond to different operating conditions. Alternatively, Chan and Kahng [36] propose tunable ROs which can be used as generic or design-dependent monitors. To obtain more accurate (design-dependent) performance estimations, the tunable ROs require calibrations at skewed process corners.

By coupling process parameters extracted from *parametric monitors* with a design-specific delay model, more accurate delay estimation can be obtained from generic test structures [28] [38] [169]. Such an approach is flexible because an arbitrary delay model can be used and calibrated at the post-manufacturing stage. Meanwhile, parametric monitors can be designed

such that they are highly sensitive to the targeted process variation. However, this approach requires a large amount of calibration and resources for storage and computation of parameters. Another class of design-dependent monitors – *in-situ monitors* [19] [73] [125] [157] [184] [204] [209] – estimates circuit performance by measuring delays of the critical paths. However, use of an in-situ monitor for each critical path incurs a high area overhead. To reduce the number of monitors, Lai et al. [125] propose to selectively measure the delays of nodes in a netlist to estimate critical path delays. Although in-situ monitors are accurate, they may increase design turnaround time because embedding in-situ monitors interferes with the timing of actual critical paths.

We propose a systematic methodology to synthesize multiple design-dependent ROs (DDROs) for circuit performance monitoring. A crucial and enabling observation is that the critical path delay sensitivities to variation sources form natural clusters (see Figure 2.3). Therefore, we can capture the design-specific delay sensitivities by synthesizing a monitor to match the delay sensitivities of each cluster. This approach has a lower implementation overhead compared to tracking each critical path because the number of clusters is much smaller than the number of the critical paths.

The potential benefits of our DDRO approach compared to the previous works are as follows.

- DDROs are more accurate compared to conventional ROs because they are synthesized to match the delay sensitivities of critical paths.
- DDROs are more accurate compared to a single RCP because multiple DDROs are used to account for the differences between critical paths.
- DDROs are less intrusive compared to in-situ monitoring methods.
- The total number of ROs (silicon area) is greatly reduced due to the clustering of critical paths. Only a few DDROs are required to provide accurate delay estimation.
- DDROs can be used for early process tuning, post-silicon tuning and real-time performance monitoring. Switching the monitoring purpose is simply a matter of redefining target variation sources (manufacturing or real-time variations) with minimal design modifications.

Since DDROs are *replica-like monitors*, they can only replicate the impact of global variation on critical paths. Thus, our monitoring approach is more suitable for long critical paths that pass

through many gates. If within-die variation dominates chip performance (e.g., chip performance is limited by hold-time critical paths and within-die variation is large), in-situ monitor is required for accurate performance estimation. Due to this inherent limitation of replica-like monitors, we only consider setup-timing critical paths.
Our contributions are summarized as follows.

- We propose a systematic methodology to design multiple DDROs. Our experimental results show that use of multiple DDROs can reduce delay overestimation by 15% to 25% compared to using only one DDRO.
- We tape out a testchip and obtain silicon measurement results showing that DDRO can reduce the mean delay estimation error by 35% compared to a generic inverter-based RO.
- We propose a method to estimate chip delay and minimize guardband margin by using multiple DDRO measurements. Our delay estimation method has negligible difference compared to a path-based estimation method, but the number of parameters used by our estimation method is significantly reduced.
- We propose a calibration method to reduce delay-estimation error due to a skewed *process, voltage and temperature* (PVT) corner.

All notations used in this paper are defined in Table 2.1.

2.1.1 Overview of DDRO Approach

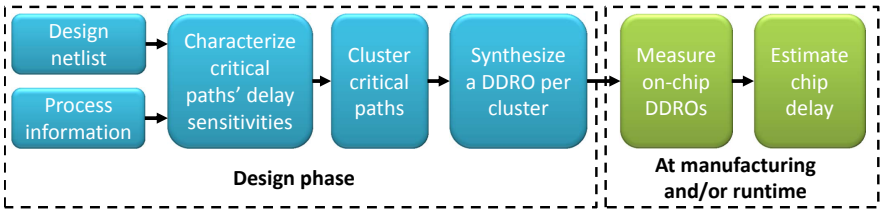


Figure 2.1: Overview of DDRO design methodology.

An overview of our monitoring strategy is shown in Figure 2.1. First, given a netlist (die area for DDROs is preallocated), we extract the critical paths of a design by running static timing analysis using both fast corner (FF) and slow corner (SS) libraries. We consider a path to be critical if its setup timing slack differs by $\leq 10\%$ of the clock period from the minimum (worst) timing slack over all paths at the corresponding process corner. For example, when the

Table 2.1: Glossary of terminology.

Term	Description
N_{mod}	Total number of gate-module types
N_{path}	Total number of critical paths
N_{var}	Total number of variation sources
N_{ro}	Total number of DDROs
N_{clust}	Total number of clusters
N_{gate}	Total number of gate instances
$d^{gate}_{nom\_h}$	Nominal delay of the h^{th} gate module
$d^{ro}_{nom\_y}$	Nominal delay of the y^{th} DDRO
$d^{clust}_{nom\_x}$	Nominal delay of the x^{th} cluster
$d^{path}_{nom\_j}$	Nominal delay of the j^{th} critical path
d^{ro}_{y}	Delay of the y^{th} DDRO
$d^{ro}_{meas\_y}$	Delay of the y^{th} DDRO measured from a chip
$d^{ro}_{meas\_y,e}$	Delay of the y^{th} DDRO measured from the e^{th} chip
d^{clust}_{x}	Delay of the x^{th} cluster
d^{path}_{j}	Delay of the j^{th} critical path
d^{chip}_{max}	Maximum delay of a chip
$d^{chip}_{max\_est}$	Estimated maximum delay of a chip
$d^{chip}_{max\_est\_j}$	Estimated maximum delay of the j^{th} chip
$d^{chip}_{max\_cal\_j}$	Estimated maximum delay of the j^{th} chip with calibration
$d^{path}_{j,v}$	Delay of the j^{th} critical path when variation source v is biased by one standard deviation
$\Delta\mathbf{D}^{ro}_{y}$	Delay sensitivities of the y^{th} DDRO to all N_{var} variation sources
$\Delta\mathbf{D}^{clust}_{max\_x}$	Delay sensitivities of the x^{th} cluster to all N_{var} variation sources
$\Delta\mathbf{D}^{path}_{j}$	Delay sensitivities of the j^{th} critical path to all N_{var} variation sources
$\Delta\mathbf{D}^{clust}_{res\_x}$	Residue of delay sensitivity in the x^{th} cluster
$\Delta\mathbf{D}^{path}_{res\_j}$	Residue of delay sensitivity of the j^{th} critical path
$\Delta\mathbf{D}^{gate}_{h}$	Delay sensitivity of the h^{th} gate module
$b_{j,y}$	Coefficient for the y^{th} DDRO after decomposing delay sensitivities of the j^{th} critical path according to delay sensitivities of DDROs
$a_{x,y}$	Coefficient for the y^{th} DDRO after decomposing delay sensitivities of the x^{th} cluster according to delay sensitivities of DDROs
$\mathbf{\Lambda}$	Correlation matrix for local variation of critical path delays
$\mathbf{g}$	Global variation vector with all N_{var} variation sources
$\mathbf{g}_e$	Global variation vector of the e^{th} chip with all N_{var} variations sources
l^{path}_{j}	Local variation of the j^{th} critical path
z'_{y}	A random variable that represents delay noise of the y^{th} DDRO
l^{tot}_{j}	Delay estimation uncertainty for the j^{th} critical path
l^{clust}_{x}	Local delay variation of the x^{th} cluster
$r_{g,z}$	A constant coefficient
z_v	Standard normal random variable
$\mathbb{E}(\cdot)$	Expectation (mean) function
$erf(\cdot)$	Error function of Gaussian distribution
$\mathbb{P}(\cdot)$	Probability function
A_{user}	User-defined confidence, $0 \leq A_{user} < 1$
A_h	Integer variable for the h^{th} gate module

design has a minimum timing slack of $10ps$ and clock period = $1ns$, paths with timing slack less than $110ps$ are considered to be critical paths. We then characterize delay sensitivities of the critical paths to variation sources using *Synopsys HSPICE* [251] with a typical (TT) corner process model.[1] Delay sensitivity of the j^{th} critical path ($\Delta\mathbf{D}_j^{path}$) is obtained by using finite differences, i.e.,

$$\Delta\mathbf{D}_j^{path} = \frac{1}{d_{nom\_j}^{path}} \left[(d_{j,1}^{path} - d_{nom\_j}^{path}), \ldots, (d_{j,N_{var}}^{path} - d_{nom\_j}^{path}) \right] \tag{2.1}$$

where $d_{j,v}^{path}$ is the delay of the j^{th} critical path when the v^{th} variation source is biased by one standard deviation from its nominal value, and $d_{nom\_j}^{path}$ is the nominal delay of the j^{th} critical path. Second, we cluster the critical paths based on their path delay sensitivities, and synthesize one DDRO per cluster. We formulate DDRO synthesis as an ILP problem, in which we seek the set of gates (gate types and number of gates of each gate type) to be concatenated as a DDRO that matches cluster delay sensitivities. Since the gate delays are sensitive to the gate capacitance and slew of adjacent gates, we use *gate modules* (i.e., several identical gates connected in series) as basic building blocks for DDRO (see Section 2.1.3). To replicate the effect of interconnect, each gate module has variants with different wirelengths (e.g., INVX1 with $5\mu m$ and $20\mu m$ wirelengths). By matching DDRO and cluster delay sensitivities, we ensure that the synthesized DDROs have good correlation with the critical paths. Since we use standard cells to synthesize the DDROs, the design and placement of DDROs can be easily integrated with conventional implementation flows. By measuring on-chip DDRO delays, we can estimate chip delay during manufacturing or runtime.

A circuit performance monitor typically feeds back the estimated delay with some margin to ensure chip functional correctness. However, the margin should be minimized to avoid significant performance overhead due to a pessimistic delay estimation. Thus, our goal for circuit performance monitoring is:

$$\begin{aligned} &\text{minimize } \mathbb{E}(d_{max\_est}^{chip} - d_{max}^{chip}) \\ &\text{subject to } \mathbb{P}(d_{max\_est}^{chip} \geq d_{max}^{chip}) > A_{user} \end{aligned} \tag{2.2}$$

where d_{max}^{chip} is the *actual chip delay*, which is defined as the maximum delay across all critical paths. Also, $d_{max\_est}^{chip}$ is the *estimated chip delay*; $\mathbb{P}(d_{max\_est}^{chip} \geq d_{max}^{chip})$ is the probability that

[1] Improved critical-path selection algorithms have been proposed in [210] [222]. Study of alternatives for path selection is beyond the scope of this thesis.

$d^{chip}_{max\_est}$ is larger than d^{chip}_{max}; and $\mathbb{E}(d^{chip}_{max\_est} - d^{chip}_{max})$ is the expectation of delay overestimation. We use A_{user} to denote a user-specified confidence. For simplicity, we call critical paths as paths in the remainder of this Section 2.1 when there is no ambiguity.

2.1.2 Delay Estimation Using DDROs

Given a set of DDROs, different chip performance estimation methods lead to different estimation errors, runtime, memory requirements, etc. We first analyze a path-based delay estimation method based on a linear model. Then, we propose a cluster-based estimation method which achieves similar accuracy but runs significantly faster and consumes less memory.

Delay and Variation Model

We use the variation model in [49], whereby lot-to-lot, wafer-to-wafer, and die-to-die process variations are lumped and modeled as global chip variation. The global variation also includes die-to-die supply voltage and temperature fluctuations. Within-die gate delay mismatches are modeled as random delay variations. Spatial variation is ignored as it is small for most chips [49]. When the effect of spatial variation is significant, DDROs can be distributed within a die as in [192] to improve correlations between DDROs and the critical paths. We model the critical path delay (d^{path}_j) as a linear function of the variation sources

$$d^{path}_j = d^{path}_{nom\_j}(1 + \Delta\mathbf{D}^{path}_j \cdot \mathbf{g} + l^{path}_j)$$

$$\begin{bmatrix} l^{path}_1 \\ \vdots \\ l^{path}_{N_{gate}} \end{bmatrix} = \mathbf{\Lambda} \cdot \begin{bmatrix} z_1 \\ \vdots \\ z_{N_{path}} \end{bmatrix} \tag{2.3}$$

where $\mathbf{g}$ is a $N_{var} \times 1$ vector that represents the global variation of N_{var} variation sources. l^{path}_j is the local delay variation of the j^{th} path. $\mathbf{\Lambda}$ is a $N_{path} \times N_{gate}$ correlation matrix that represents the correlation between paths, where N_{path} is the total number of paths, and N_{gate} is the total number of gate instances in all N_{path} paths. $z_1, ..., z_{N_{path}}$ are independent random variables, each of which follows a standard normal distribution.[2]

[2]We obtain $\mathbf{\Lambda}$ by running SPICE simulations [245] with a variation model that is embedded in the foundry PDK for the $45nm$ SOI process.

To verify the accuracy of our delay model, we first simulate a critical path using *Synopsys HSPICE* [251] with random global variations whose sources are as listed in Table 2.2 (100 trials). Then, we compare the simulated path delays with the delays calculated using the linear model in Equation (2.3). Figure 2.2 shows that path delays obtained from the linear model correlate very well with those from the SPICE simulations.

Since a RO has many identical gates, uncorrelated local variation is insignificant due to averaging of uncorrelated delay variation. Therefore, we do not model local variation in the DDROs, i.e., we use

$$d_y^{ro} = d_{nom\_y}^{ro}(1 + \Delta\mathbf{D}_y^{ro} \cdot \mathbf{g}) \tag{2.4}$$

where d_y^{ro} is the delay of the y^{th} RO, $d_{nom\_y}^{ro}$ is the nominal delay of the y^{th} DDRO (obtained from simulation) and $\Delta\mathbf{D}_y^{ro}$ is a $1\times N_{var}$ vector that represents the delay sensitivity of the y^{th} DDRO to the vector $\mathbf{g}$ of all N_{var} global process variations.

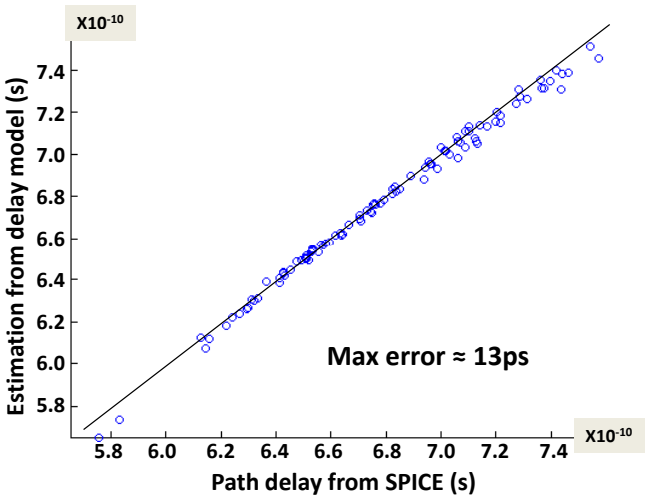


Figure 2.2: Rank correlation between delays obtained from SPICE simulation and the linear model of Equation (2.3).

Path-Based Delay Estimation

A straightforward delay estimation method is to extract global variation using multiple process variation-specific monitors and calculate chip delay based on the linear model in Equation (2.3). In other words, monitoring methods in [28] [38] and [169] can be combined and extended for delay estimation. However, we use this approach only as a reference because it requires a large amount of memory to store parameters, as well as long computation time.

Given N_{ro} DDROs, we can decompose the vector of delay sensitivities $\Delta\mathbf{D}_j^{path}$ as a linear combination of $\Delta\mathbf{D}_y^{ro}$ ($y = 1, ..., N_{ro}$) to utilize measurements from the DDROs.

$$\Delta\mathbf{D}_j^{path} = \sum_{y=1}^{Y} b_{j,y} \cdot \Delta\mathbf{D}_y^{ro} + \Delta\mathbf{D}_{res\_j}^{path} \tag{2.5}$$

where $b_{j,y}$ is a constant coefficient and $\Delta\mathbf{D}_{res\_j}^{path}$ is a $1 \times N_{var}$ vector that represents the residue of the delay-sensitivity decomposition.[3] The values of $b_{j,y}$ are obtained by solving a linear program (see Section 2.1.3). Substituting $\Delta\mathbf{D}_j^{path}$ in Equation (2.3) as a linear combination of $\Delta\mathbf{D}_y^{ro}$, we obtain

$$\begin{aligned} d_j^{path} &= d_{nom\_j}^{path}(1 + \sum_{y=1}^{N_{ro}} \overbrace{(b_{j,y} \cdot \Delta\mathbf{D}_y^{ro} \cdot \mathbf{g})}^{\text{measurable}} + \overbrace{l_j^{tot}}^{\text{uncertainty}}) \\ \text{where } l_j^{tot} &= l_j^{path} + \Delta\mathbf{D}_{res\_j}^{path} \cdot \mathbf{g} \end{aligned} \tag{2.6}$$

Equation (2.6) shows that d_j^{path} consists of a measurable term and an uncertainty term. While the value of the measurable term can be determined from the delays of DDROs, the value of the uncertainty term cannot be measured directly. To estimate the maximum chip delay with the uncertainty l_j^{tot}, we calculate the distribution of the chip maximum frequency, d_{max}^{chip}, by using the method in [202]. Then, we can express d_{max}^{chip} as a normal distribution using a mean $\mathbb{E}(d_{max}^{chip})$ and a standard deviation $\sigma(d_{max}^{chip})$. Also, the $d_{max\_est}^{chip}$ can be readily obtained using the *erf* function for Gaussian distribution.

$$erf(\frac{\sigma(d_{max}^{chip}) - \mathbb{E}(d_{max}^{chip})}{\sigma(d_{max}^{chip})}) > A_{user} \tag{2.7}$$

Clustering

The next step is to minimize delay margin and find $\Delta\mathbf{D}_y^{ro}$. Equations (2.6) and (2.7) show that a larger $\Delta\mathbf{D}_{res\_j}^{path}$ will lead to a larger $d_{max\_est}^{chip}$. Therefore, it is desirable to select a set of $\Delta\mathbf{D}_k^{ro}$ that minimizes $\Delta\mathbf{D}_{res\_j}^{path}$. To address this problem, we find $\Delta\mathbf{D}_k^{ro}$ by clustering critical paths with similar $\Delta\mathbf{D}_j^{path}$ sensitivity vectors we then assign the centroid of the x^{th} cluster as

[3] Since there will be no residue when $N_{ro} = N_{var}$, it is preferred to have $N_{ro} < N_{var}$. We try $N_{ro} = \{1, 3, 5, 7, 12\}$ and show that $N_{ro} = 5$ is sufficient for our testcases with 12 variation sources ($N_{var} = 12$).

$\Delta\mathbf{D}_x^{ro}$. To cluster the paths, we use the *kmeans*++ algorithm [7] and choose the best clustering solution among 100 random starts. The objective function of the clustering is defined as

$$\begin{aligned}\text{minimize} \sum_{j=1}^{N_{path}} \{&\mathbb{P}(d_j^{path} > \text{clock period}) \\ &\times ||\Delta\mathbf{D}_x^{ro} - \Delta\mathbf{D}_j^{path}||\},\ \text{path } j \in \text{cluster } x\end{aligned} \tag{2.8}$$

Since the maximum chip delay is usually determined by the slowest path, we impose a higher penalty for having mismatched delay sensitivities on a path with higher probability of timing failure, i.e., $\mathbb{P}(d_j^{path} > \text{clock period})$. For each path, the probability of timing failure is calculated based on the delay model in Equation (2.3) and the distributions of variation sources, $\mathbf{g}$. Minimizing the cost function in Equation (2.8) helps reduce the upper bound of $\Delta\mathbf{D}_{res\_j}^{path}$ because the upper bound is defined by $\Delta\mathbf{D}_x^{ro} - \Delta\mathbf{D}_j^{path}$. An example clustering result is shown in Figure 2.3.

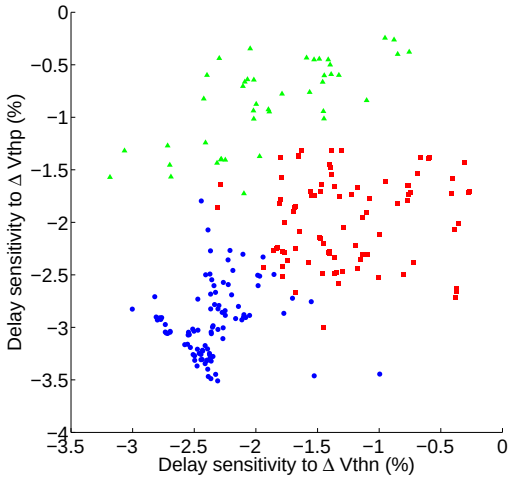


Figure 2.3: Every dot in the figure represents a critical path's delay deviation for one standard deviation in NMOS threshold voltage (V_{thn}) and PMOS threshold voltage (V_{thp}). We cluster the paths into three clusters (according to all 12 variation sources) and indicate the 3-way clustering by different marks.

Cluster-Based Delay Estimation

The path-based delay estimation method requires $O(N_{path} \cdot N_{gate})$ parameters for run-time delay estimation. To reduce the number of parameters, we represent path delays in a cluster by the maximum path delays within a cluster (d_x^{clust}). We calculate the maximum delay of paths

in each cluster using the method in [202] and the path delay model of Equation (2.3). The outcome of this step gives us the expected maximum delay of cluster x. But more importantly, it also extracts the sensitivities of the maximum delay to variation sources ($\Delta\mathbf{D}^{clust}_{max\_x}$). Similar to the path-based approach, we represent $\Delta\mathbf{D}^{clust}_{max\_x}$ as a function of $\Delta\mathbf{D}^{ro}_{y}$:

$$\Delta\mathbf{D}^{clust}_{max\_x} = \sum_{y=1}^{N_{ro}} \{a_{x,y} \cdot \Delta\mathbf{D}^{ro}_{y}\} + \Delta\mathbf{D}^{clust}_{res\_x} \tag{2.9}$$

where $a_{x,y}$ is a constant coefficient, and $\Delta\mathbf{D}^{clust}_{res\_x}$ is the residue of the delay-sensitivity decomposition. Note that when $\Delta\mathbf{D}^{ro}_{y}$ is equal to $\Delta\mathbf{D}^{clust}_{max\_x}$, $\Delta\mathbf{D}^{clust}_{res\_x} = 0$. However, the synthesized $\Delta\mathbf{D}^{ro}_{y}$ are usually slightly different from $\Delta\mathbf{D}^{clust}_{max\_x}$. Thus, having $a_{x,y}$ is useful to reduce $\Delta\mathbf{D}^{clust}_{res\_x}$. The approximate delay of the x^{th} cluster is given by

$$\begin{aligned} d^{clust}_{x} = d^{clust}_{nom\_x}(1 + \sum_{y=1}^{N_{ro}}(a_{x,y} \cdot \Delta\mathbf{D}^{ro}_{y} \cdot \mathbf{g}) \\ + \Delta\mathbf{D}^{clust}_{res\_x} \cdot \mathbf{g} + l^{clust}_{x}) \end{aligned} \tag{2.10}$$

where d^{clust}_{x} denotes the delay of the x^{th} cluster, $d^{clust}_{nom\_x}$ represents the nominal delay of the x^{th} cluster, and l^{clust}_{x} represents the random local delay of the x^{th} cluster. After measuring DDROs, we can obtain the mean and standard deviation of d^{clust}_{x} as in Equation (2.11).

$$\begin{aligned} \sigma(d^{clust}_{x}) &= \{\sigma(||\Delta\mathbf{D}^{clust}_{res\_x} \cdot \mathbf{g}||)^2 + \sigma(l^{clust}_{x})^2\}^{\frac{1}{2}} \\ \mathbb{E}(d^{clust}_{x}) &= d^{clust}_{nom\_x}(1 + \sum_{y=1}^{N_{ro}}(a_{x,y} \cdot \Delta\mathbf{D}^{ro}_{y} \cdot \mathbf{g})) \end{aligned} \tag{2.11}$$

Then, we can calculate the maximum delay distribution of a chip, d^{chip}_{max}, using the method in [202] and find the value of $d^{chip}_{max\_est}$ using Equation (2.7). Although the number of clusters need not be the same as N_{ro}, we let each cluster correspond to one DDRO. Using this cluster-based approximation method consumes less memory compared to the path-based method because the total number of parameters is reduced from $O(N_{path} \cdot N_{gate})$ to $O(N^2_{ro})$, where $N_{ro} \ll N_{path} \ll N_{gate}$. Moreover, the number of operations to calculate the maximum of two delay distributions is reduced from $O(N_{path})$ to $O(N_{ro})$. This reduces maximum-delay calculation time from a minute (with the path-based method) to less than a second (with the

cluster-based method).[4] The cluster-based (fast) delay estimation method could enable the use of DDROs for real-time performance monitoring, which requires monitors to feed back chip performance variation (due to temperature or voltage variation) as soon as possible such that the chip can adapt to the variations. When DDROs are used for post-silicon tuning, the cluster-based delay estimation method can reduce calibration time.

2.1.3 Synthesis of DDROs

Given a delay sensitivity target ($\Delta\mathbf{D}_y^{ro}$), we want to construct a DDRO such that the delay sensitivities of the DDRO match the targeted delay sensitivities. This DDRO synthesis problem is difficult because there can be many combinations of gates to construct a RO. Here, we describe an ILP formulation to solve the DDRO synthesis problem. Further, we describe various aspects which must be considered during DDRO synthesis.

ILP Formulation

Since each gate-module type is instantiated a discrete number of times, we formulate DDRO synthesis as an ILP problem:

$$\begin{aligned} \text{minimize} & \Big| \sum_{h=1}^{N_{mod}} \{d_{nom\_h}^{gate} \times A_h\} \times \Delta\mathbf{D}^{ro} - \sum_{h=1}^{N_{mod}} \{d_{nom\_h}^{gate} \times A_h \times \Delta\mathbf{D}_h^{gate}\} \Big| \\ \text{subject to} & \sum_{h=1}^{N_{mod}} d_{nom\_h}^{gate} \times A_h \ \geq \ \text{minimum DDRO delay} \\ & \sum_{h=1}^{N_{mod}} A_h \qquad\qquad \leq \ \text{maximum gate count} \end{aligned} \tag{2.12}$$

where $d_{nom\_h}^{gate}$ is the nominal delay of candidate gate-module type h and A_h is the integer variable that indicates the number of copies of gate-module type h in the DDRO. $\Delta\mathbf{D}_h^{gate}\}$ is delay sensitivities to all N_{var} variation sources for the h^{th} gate module. N_{mod} is the total number of gate-module types. After solving the ILP, $|A_h|$ copies of gate-module type h are used in the DDRO. If $|A_h|$ is zero, gate-module type h is not used in the DDRO. In our experiments, solving the ILP with the *LP_SOLVE* solver [238] takes one hour on a $3GHz$ single-core CPU. Instead of minimizing the difference in relative delay sensitivity, the formulation in Equation (2.12) minimizes the absolute delay-sensitivity difference such that the objective function is linear in

[4] In our experiment, calculating the maximum delay distribution of several hundreds of paths with a $3GHz$ single-core CPU takes up to a minute of CPU time.

A_h. This favors a solution with a smaller DDRO nominal delay, which may be suboptimal in term of normalized delay-sensitivity difference. To compensate this inherent bias in the ILP, we add a constraint to define a minimum allowed DDRO delay. We then sweep the value of minimum DDRO delay at across evenly-spaced values within its feasible range.

Selecting Major Variation Sources

Table 2.2: List of variation sources.

Parameter	Descriptions
V_{dd}	Supply voltage. V_{dd} nominal (V_{nom}) is $0.9V$, $3\sigma = 0.05 \times V_{nom}$ = $45mV$.
Temperature	Ambient temperature. Nominal temperature = 25°C, 3σ = 30°C.
C_{gdo}	MOSFET gate overlap capacitance at drain junction
C_{gso}	MOSFET gate overlap capacitance at source junction
R_{dsw}	Channel series resistance per unit width
μ_0	Mobility of MOSFET
L_{gate}	MOSFET gate length
T_{ox}	Oxide thickness of MOSFET
V_{thn_r}	Threshold voltage of RVT NMOS
V_{thp_r}	Threshold voltage of RVT PMOS
V_{vtn_h}	Threshold voltage of HVT NMOS
V_{thp_h}	Threshold voltage of HVT PMOS

To identify major variation sources that affect delay sensitivity, we simulate a seven-stage RO using the foundry-supplied $45nm$ SOI SPICE model. The SPICE model has 13 process-related parameters for process variation analysis. In our experiment, we perturb all of these 13 parameters (one at a time), as well as the supply voltage and temperature. Based on the results in Figure 2.4, we can see that most of the variation sources have noticeable effect on the delay except for C_{gdl}, C_{gsl} and C_{jswg}. Therefore, we only consider 12 out of the 15 major variation sources; these are summarized in Table 2.2.[5] We do not include second-order sensitivities to the variation sources because their magnitudes are very small. This assumption is supported by the data in Figure 2.2.

In our experimental setup, the impact of interconnect is modeled by parasitic resistance and capacitance extracted from design layout. However, we do not model interconnect as a

[5]Unless otherwise mentioned, the σ values of variation sources are taken from the foundry $45nm$ SOI process.

variation source because its impact is relatively small compared to that of active devices [32]. If interconnect variations are to be included, the DDRO must be built with components that are sensitive to interconnect variations.

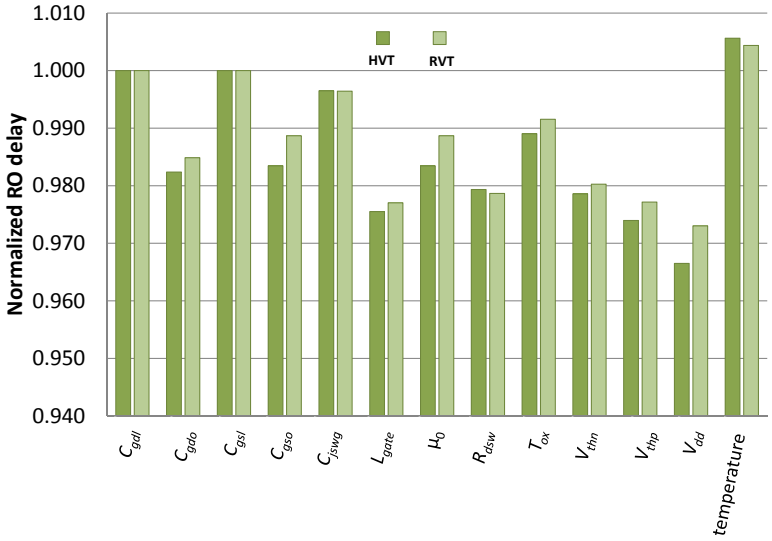


Figure 2.4: Delay sensitivities of an RO to different variation sources show that most of the sources have noticeable effect except for C_{gdl}, C_{gsl} and C_{jswg}. Delays (y-axis) are normalized with respect to the nominal delay of the RO with no variation.

Characterizing Gate Sensitivities

Our ILP formulation in (2.12) assumes that delay sensitivity of a gate (standard cell) is not sensitive to other gates connected before and after it. This is a key assumption that simplifies the problem. If we model $\Delta \mathbf{D}_h^{gate}$ as a function of its adjacent gate type, the total number of variables and the design space become intractable.

To decouple the load and slew interaction between the gates, we introduce gate modules as basic building blocks for DDRO. A gate module is defined as several identical gates connected in series as illustrated in Figure 2.5. Simulation results in Figure 2.6 show that the sensitivity difference due to different input slew and output load is reduced from 0.15% to 0.03%, as the number of stages in a gate module increases from 1 to 15. We use five-stage gate modules as a result of tradeoff between stability of sensitivity and total area of a gate module.

For a gate with multiple input pins, gate delays through different input pins will have different delay sensitivities. Thus, each gate-module type is defined with respect to a specific input pin. For example, gate-module types NAND2X1_A and NAND2X1_B use the same gate type

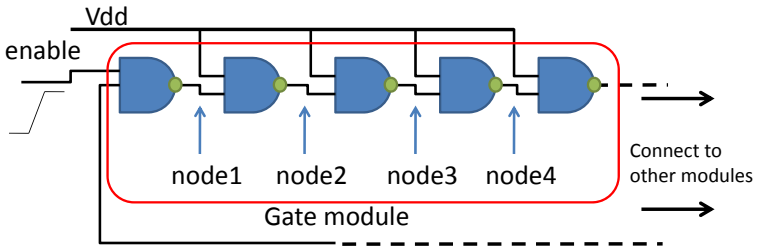


Figure 2.5: Illustration of a gate module in a DDRO.

(NANDX1) but the gate modules toggle different input pins. Extra input pins of a multi-input gate are assigned to high or low to make a gate module inverting or buffering (see Figure 2.5). To obtain a list of candidate gate-module types for DDRO synthesis, we use logic standard cells (e.g., AND, OR, XOR, INV gates) to build gate modules. For multi-input gates, we generate a gate-module type for each input pin. Since there are many gate-module types, we select those which have similar gate capacitance. This is because gate modules with similar gate capacitance have less impact on the delay sensitivities of adjacent gate modules when they are concatenated to form a DDRO.

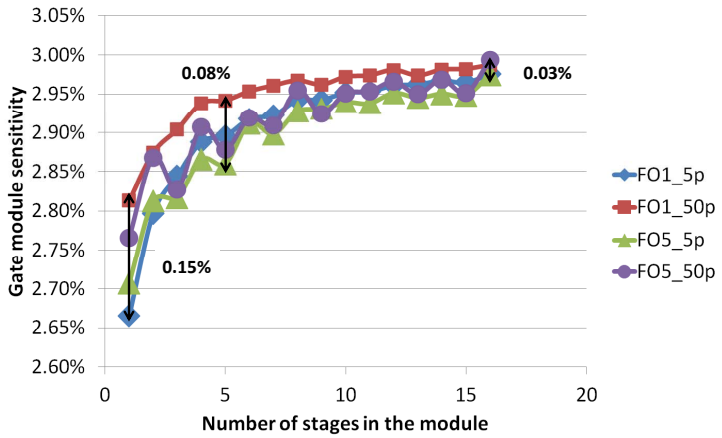


Figure 2.6: Simulation results show that the sensitivities under different input slew $\{5ps$, $50ps\}$ and output load {FO1, FO5} combinations converge as the number of stages in a gate module increases.

Since the interconnect also affects path delay sensitivity, we use different wirelengths in building our gate modules. Gate modules with different wirelengths are considered as different instance types even if they have the same gate type. Note that the gate-module wirelengths need to be defined based on both the technology and the critical paths that are to be monitored. In our experiment, the wirelengths of critical paths are typically less than $20\mu m$ (see Figure 2.7).

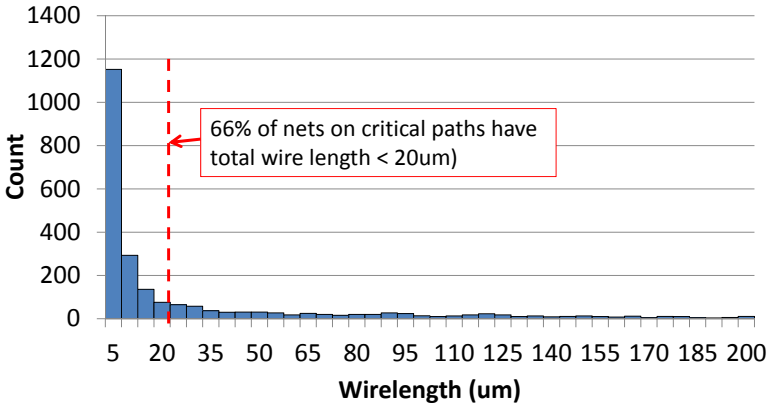


Figure 2.7: Wirelength distribution of each net on critical paths. The critical paths are extracted from an ARM *Cortex-M3* processor implemented using a foundry $45nm$ SOI technology.

Thus, we use two types of interconnect lengths in our gate modules, i.e., the wirelength between consecutive gates in a module can be either short ($5\mu m$) or long ($20\mu m$). As depicted in Figure 2.8, we create custom *interconnect cells* with "snaking" routes to match the desired interconnect wirelengths as well as reduce the total area of DDROs. During physical implementation, we synthesize each DDRO using gate modules which consist of standard cells and custom interconnect cells. The gates modules in each DDRO are placed in two rows to form a loop. The standard cells and the custom interconnect cells in each gate module are placed in series.

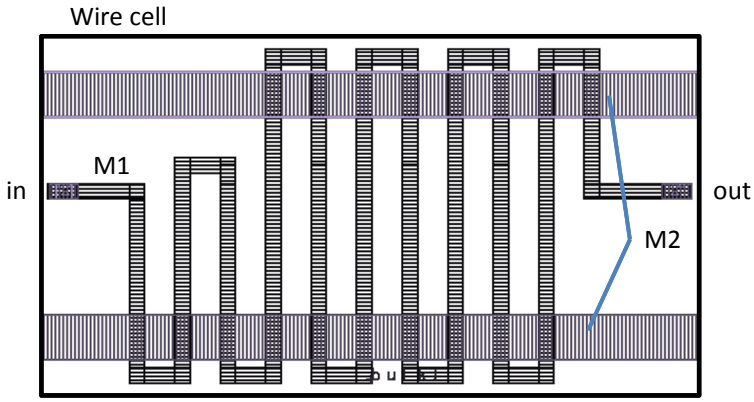


Figure 2.8: Custom interconnect cell with a snaking route to reduce total area of long interconnect.

Extraction of $b_{j,y}$ And $a_{x,y}$

As mentioned in Section 2.1.2, we represent $\Delta\mathbf{D}_j^{path}$ and $\Delta\mathbf{D}_{max\_x}^{clust}$ as linear combinations of $\Delta\mathbf{D}_y^{ro}$, using $b_{j,y}$ and $a_{x,y}$, respectively. The $b_{j,y}$ (resp. $a_{x,y}$) extraction is achieved by solving Equation (2.5) (resp. Equation (2.9)) using simple least-squares fitting to minimize the resulting residue, $\Delta\mathbf{D}_{res\_j}^{path}$ (resp. $\Delta\mathbf{D}_{res\_x}^{clust}$). However, the simple fitting approach can lead to overfitting when $N_{ro} \approx N_{var}$, which results in large $b_{j,y}$ (resp. $a_{x,y}$) values and increases delay-estimation error. For example, Figure 2.9(a) (left) shows that solving Equation (2.5) using a *linear least-squares* method without constraints on $b_{j,y}$ leads to little delay overestimation when we consider global variation only. However, Figure 2.9(a) (right) shows that this is not true when we repeat the experiment with global and local variations, as well as other variations that are absent in our delay model. This is because the large $b_{j,y}$ (resp. $a_{x,y}$) values magnify *delay noise*, i.e., the differences between the actual delays and the delays calculated using the linear delay model in Equation (2.3). The delay noise is mainly due to the fact that critical path and DDRO delays have nonlinear dependence on parameters in Table 2.2, when subjected to PVT variations.

To reduce the impact of large $b_{j,y}$ (resp. $a_{x,y}$) values, [34] formulates the extraction problem as a linear program with upper and lower bounds on $b_{j,y}$ (resp. $a_{x,y}$). Although the method of [34] avoids large estimation error, the upper and lower bounds are determined by trial-and-error to minimize delay-estimation error.

We consider both RO delay-sensitivity decomposition residue and delay noise as errors and formulate the $b_{j,y}$ (resp. $a_{x,y}$) extraction problem as a linear program.

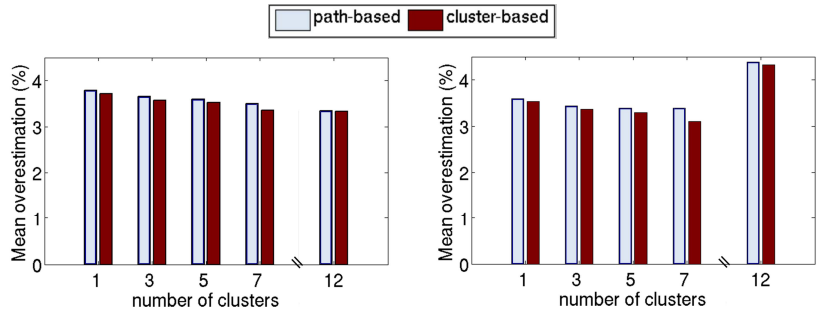
$$\text{minimize} \quad \Delta\mathbf{D}_{res\_j}^{path} \cdot \mathbf{g} + [b_{j,1} \ \ldots \ b_{j,N_{ro}}] \cdot \begin{bmatrix} z'_1 \\ \vdots \\ z'_{N_{ro}} \end{bmatrix} \tag{2.13}$$

where z'_y is a random variable that represents the delay noise of DDRO y introduced by the linear delay approximation in Equation (2.4). Note that the z'_y also includes higher-order delay sensitivities, any unmodeled variation, as well as the local variation in DDRO due to process variations.

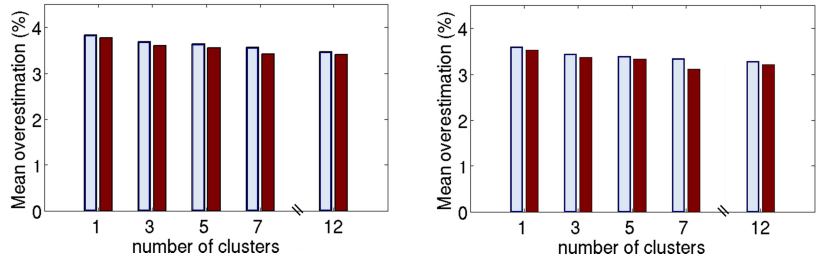
The value of z'_y can be estimated by calculating the difference of the delay obtained from SPICE Monte Carlo simulation and that from Equation (2.4). Alternatively, we can define $r_{g,z}$

as the ratio between $\mathbf{g}$ and z'_y and simplify the linear program in (2.13) as

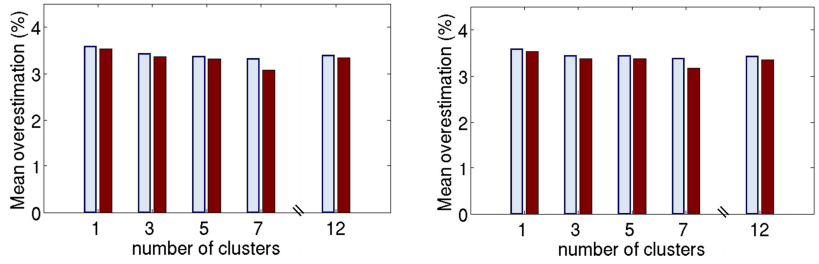
$$\text{minimize} \quad \|\Delta \mathbf{D}^{path}_{res\_j}\|_2 + r_{g,z} \cdot \left|\left|\left[b_{j,1} \ \ldots \ b_{j,N_{ro}} \right]\right|\right|_2 \tag{2.14}$$



(a) Linear model results (left) versus SPICE results (right) using linear least-square method on $b_{j,y}$ for *MIPS* testcase. Linear least-square method works for linear model but becomes unstable with SPICE results.



(b) Linear model results (left) versus SPICE results (right) using our method for *MIPS* testcase with $r_{g,z} = 0.02$. With our method, the results are consistent for both linear model and SPICE results.



(c) SPICE model results with (left) $r_{g,z} = 0.01$ and (right) $r_{g,z} = 0.1$. Our method is robust and insensitive to the value of $r_{g,z}$.

Figure 2.9: Estimation error of a testcase (*MIPS*) with different setups.

Based on our empirical results, we set $r_{g,z} = 0.02$. Results in Figure 2.9(b) show that by using $a_{x,y}$ extracted by solving the problem in (2.14), the delay estimations are not sensitive to delay noise caused by circuit nonlinearity and other variations. Moreover, Figure 2.9(c) shows that the delay-estimation errors are not sensitive to $r_{g,z}$. Thus, the formulation in (2.14) is more robust than that in [34].

Synthesis Results

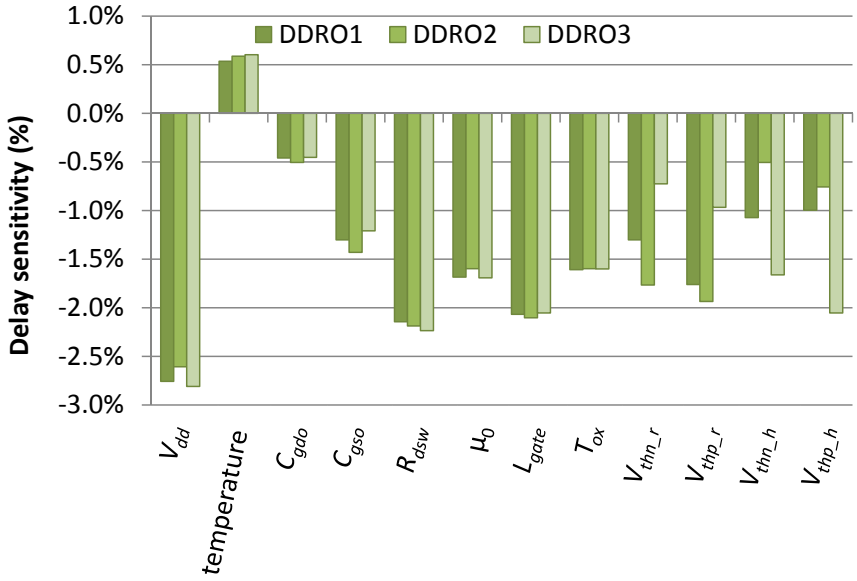


Figure 2.10: Delay sensitivities of synthesized DDROs of testcase *Cortex-M0*. Cluster number = 3. The delay sensitivities (y-axis) is normalized to DDRO delay with no variation.

Figure 2.10 shows examples of synthesized DDROs for testcase *Cortex-M0* with $N_{ro} = 3$. As shown in the figure, the synthesized DDROs have three sets of linearly-independent delay sensitivities. This is an important property because we will use linear combinations of the delay sensitivities to match the delay sensitivities of critical paths or path clusters (DDROs with linearly-dependent delay sensitivities are redundant).

Figure 2.11 shows that by using linear combinations of delay sensitivities of DDROs (i.e., $a_{x,y} \cdot \Delta\mathbf{D}_y^{ro}$), we can achieve smaller delay-sensitivity errors with respect to a critical path compared to using DDROs directly or simple inverter-based ROs. The standard cells in the DDROs are described in Table 2.3.

Delay Estimation with Skewed PVT Corner

The estimation methods in Section 2.1.2 assume that the nominal RO delays ($d_{nom\_y}^{ro}$) are obtained from SPICE simulation at the nominal PVT corner, i.e., the measurable term in Equation (2.6) is defined as

$$\Delta\mathbf{D}_y^{ro} \cdot \mathbf{g}_e = \frac{d_{meas\_y,e}^{ro}}{d_{nom\_y}^{ro}} - 1 \tag{2.15}$$

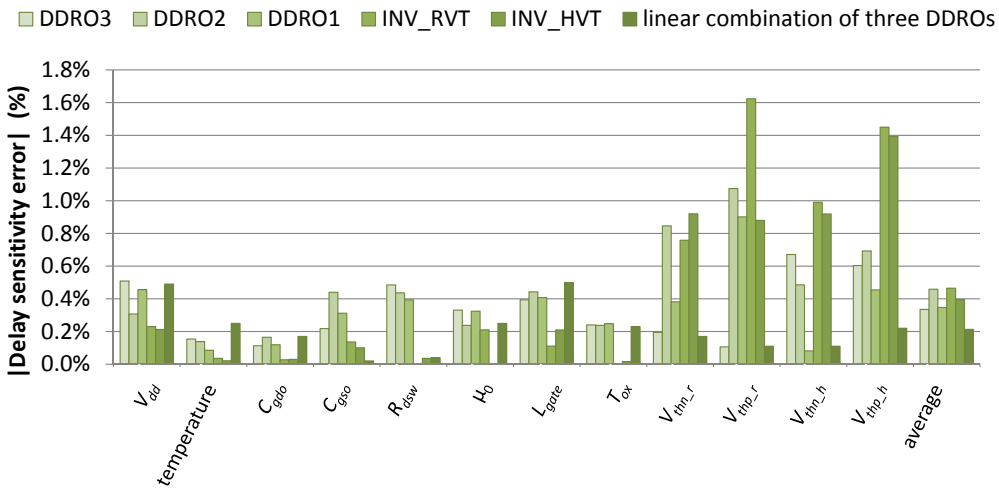


Figure 2.11: Delay-sensitivity errors of different ROs with respect to the delay sensitivities of a critical path in testcase *Cortex-M0*.

where $d^{ro}_{meas\_y,e}$ is the delay of the y^{th} DDRO measured from the e^{th} chip and $\mathbf{g}_e$ is the global process variation of the e^{th} chip. If the actual operating PVT corner of the chips is significantly skewed compared to the nominal corner, $d^{path}_{nom\_j}$ and $d^{ro}_{nom\_y}$ obtained from SPICE simulation will be inaccurate. This is especially important for low-volume production runs. Therefore, we propose a method to calibrate $d^{path}_{nom\_j}$ and $d^{ro}_{nom\_y}$ when chip samples are available. Given a set of chip samples, we can obtain the mean RO delay across all samples ($\mathbb{E}(d^{ro}_{meas\_y})$). By replacing the $d^{ro}_{nom\_y}$ in Equation (2.15) with $\mathbb{E}(d^{ro}_{meas\_y})$, we compensate for the error caused by a skewed process and/or mismatch between SPICE model and silicon data.

$$\Delta\mathbf{D}^{ro}_{y} \cdot \mathbf{g}_e = \frac{d^{ro}_{meas\_y,e}}{\mu(d^{ro}_{meas\_y})} - 1 \tag{2.16}$$

After applying the calibration in Equation (2.16), we can estimate the delay of the e^{th} chip ($d^{chip}_{max\_est\_e}$) using Equation (2.11). Similarly, the chip delay is also susceptible to the skewed process as well as mismatch between SPICE model and silicon data. Moreover, chip delay can be skewed differently with respect to the DDRO. To minimize delay-estimation error resulting from the systematic mismatch between chip and DDRO delays, we propose to apply an additional calibration procedure during chip delay estimation. First, we obtain the *expectation of actual chip delay* ($\mathbb{E}(d^{chip}_{max})$) by calculating the average of sample chip delays. Second, we calculate the *expectation of chip delay estimation* ($\mathbb{E}(d^{chip}_{max\_est})$) by averaging chip delay estimations ($d^{chip}_{max\_est\_e}$) across all chip samples. In other words, ($\mathbb{E}(d^{chip}_{max\_est})$) is defined as the average

of the expectation of estimated chip delay.

$$\mathbb{E}(d^{chip}_{max\_est}) = \frac{1}{\text{total samples}} \sum_{e} (d^{chip}_{max\_est\_e}|_{A_{user}=50\%}) \tag{2.17}$$

The *calibrated maximum-delay estimate* for chip e ($d^{chip}_{max\_cal\_e}$) is given by

$$d^{chip}_{max\_cal\_e} = \frac{\mathbb{E}(d^{chip}_{max})}{\mathbb{E}(d^{chip}_{max\_est})} \cdot d^{chip}_{max\_est\_e} \tag{2.18}$$

Table 2.3: Standard cells in DDROs.

	Copies	Wirelength	Cell type	Size	V_{th}
	5	w20	NAND2	X1.4	RVT
	5	w20	AOI222	X1.4	HVT
	20	w20	AOI31	X1.4	RVT
	10	w20	INV	X1.2	RVT
DDRO1	5	w20	OAI31	X2	HVT
	5	w20	OAI31	X3	HVT
	5	w20	OAI31	X3	HVT
	5	w20	XOR2	X1.4	RVT
	5	w5	OAI2XB1	X1.4	RVT
	5	w5	OAI31	X3	HVT
	5	w20	NAND2	X1.4	RVT
	10	w20	AOI31	X1.4	RVT
	5	w20	XNOR2	X0.5	HVT
DDRO2	20	w5	AOI31	X1.4	RVT
	5	w5	OAI221	X1.4	HVT
	15	w5	OAI31	X2	RVT
	10	w5	OAI31	X3	RVT
	5	w20	NAND2	X1.4	RVT
	15	w20	AOI221	X1.4	RVT
	5	w20	OA21A1OI2	X1.4	HVT
	5	w20	OAI211	X1.4	HVT
	10	w20	OAI222	X1.4	HVT
DDRO3	5	w20	OAI222	X1.4	RVT
	5	w20	OAI31	X2	HVT
	5	w20	OAI31	X2	RVT
	5	w20	XNOR2	X0.7	HVT
	15	w20	XOR2	X0.5	RVT
	25	w20	XOR3	X0.5	HVT

2.1.4 Experimental Results

To validate our performance-monitoring methodology, we synthesized, placed and routed three benchmark circuits using a foundry $45nm$ SOI technology. Details of the implemented benchmark designs are listed in Table 2.4. The benchmark circuits are obtained from *ARM* [224] and *OpenCores* [243]. Then, we follow the DDRO design flow in Figure 2.1. We first run static timing analysis using both FF and SS libraries. As mentioned in Section 2.1.1, we consider a path to be critical if its setup timing slack at either FF or SS corner differs from the worst timing slack at the corresponding process corner by no more than 10% of the clock period. We extract delay sensitivity of each critical path to each of the variation sources in Table 2.2 using SPICE, and a typical process model. Note that SPICE-based sensitivity characterization is not mandatory in our design flow, and that it can be replaced by other methods (e.g., the statistical method in [209]).

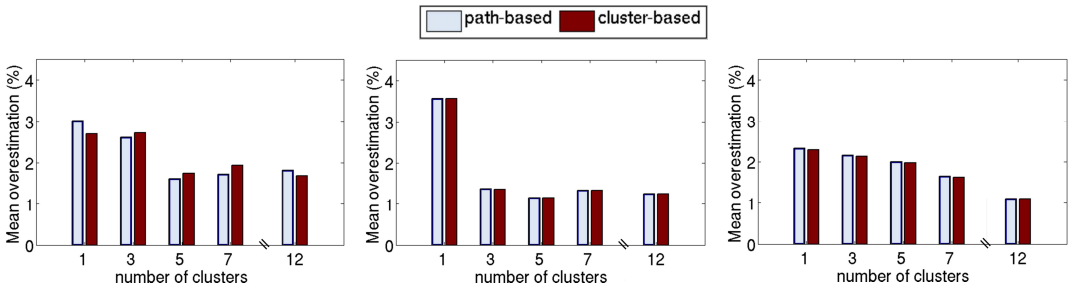
Table 2.4: Physical implementation results of benchmark circuits.

Benchmark circuit	Number of cells	Clock period	Number of critical paths
Cortex-M0	8169	1000ps	218
MIPS	8283	900ps	107
AES	10634	800ps	420

To evaluate the quality of our DDRO synthesis and delay estimation methodologies, we run Monte Carlo experiments with global and local variations on the critical paths and DDROs. For SPICE simulation, we use the built-in Monte Carlo setup in the $45nm$ SOI device model. Since each critical path is defined for a specific input and simulated independently, we cannot capture the correlation of local variation due to gate sharing among the critical paths. As an alternative, we run another set of Monte Carlo experiments using the linear model in Equation (2.3). In both simulations, we use the path and DDRO delay sensitivities extracted from SPICE simulation results to minimize the discrepancy between them. In the linear model experiment, we sample the values of variation sources by using the Gaussian random number generator in *Matlab* [239]. The number of trials in the Monte Carlo experiment is 1000 and 100 for the linear model and for SPICE simulation, respectively. Unless otherwise specified, we set the user-specified confidence A_{user} = 99%.[6]

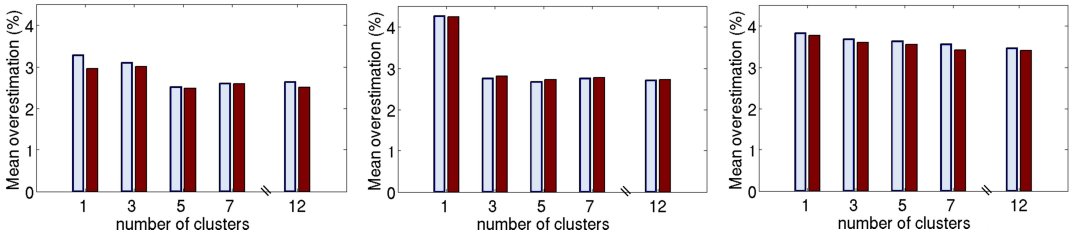
[6]When the number of trials is small, our delay estimation is more sensitive to the instances of the trials, especially for a high confidence A_{user} = 99%.

Simulation Results



(a) *AES* (b) *Cortex-M0* (c) *MIPS*

Figure 2.12: Linear model simulation results with global variations only.



(a) *AES* (b) *Cortex-M0* (c) *MIPS*

Figure 2.13: Linear model simulation results with global and local variations.

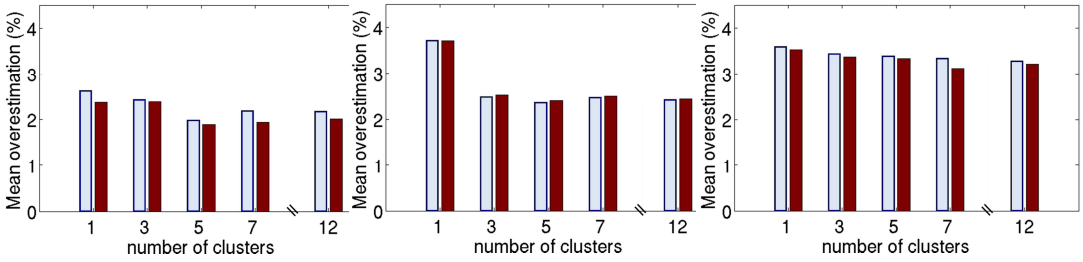
Experiments using linear model. The simulation results in Figures 2.12 and 2.13 show that our approximate delay estimation method achieves similar results compared to the path-based method. The results also show that mean delay overestimation of all benchmark circuits reduces noticeably as the number of clusters increases from 1 to 12.[7] This confirms our hypothesis that having multiple DDROs that correlate well with the critical paths can reduce chip delay overestimation. The results also show that delay overestimation is nonzero even when the number of DDROs = 12 (i.e., $N_{ro} = N_{var} = 12$). This is because $\Delta\mathbf{D}_{res\_j}^{path}$ and $\Delta\mathbf{D}_{x}^{clust}$ are nonzero.

We further observe that the benefit of using multiple DDROs is more significant when the local variation is relatively less compared to the global variation. This is because replica-like monitors (e.g., PSRO, DDRO, PLL) can only replicate the impact of global variation on the critical paths. If local variation dominates, more intrusive monitoring is required to measure the impact of local variation. Based on the simulation results with global and local variations (Figure 2.13), minimum values of delay overestimations for the *AES*, *Cortex-M0* and *MIPS* testcases are 2.5%, 2.7% and 3.4%, respectively. The results for $N_{ro} = 12$ in Figures 2.12 and

[7] When the number of clusters (N_{ro}) = 1, our DDRO method is similar to the RCP method in [137].

2.13 show that the achievable minimum delay overestimation is limited by the local variation of a design. Therefore our performance-monitoring method may be more suited for low-speed designs with longer critical paths that are less susceptible to local delay variations.

SPICE simulations. SPICE



(a) *AES* (b) *Cortex-M0* (c) *MIPS*

Figure 2.14: SPICE results for global and local variations.

Results in Figure 2.14 are similar to the linear model results. Discrepancies between SPICE and linear model results are mainly due to the fact that our delay estimation does not account for nonlinearity in circuit delay. Despite a user-specified confidence of 99%, the results in Table 2.5 show that we underestimate the delays of 1.96% and 5.9% instances in the linear model and SPICE experiments, respectively (average across three benchmarks for cluster-based estimation). Since the results of the linear model experiment are free from nonlinearity error, the underestimation error is mainly due to the approximation in the *statistical maximum function* given by [202]. The SPICE results have more underestimated instances because local variation is not modeled correctly, i.e., SPICE simulates the critical paths with uncorrelated local random variation but our delay estimation accounts for correlation between local variations. As a result, our delay estimates are slightly smaller than the path delays obtained from SPICE simulation.

Table 2.5: Average underestimated instances across $N_{ro} = \{1, 3, 5, 7, 12\}$.

Benchmark	Linear model Global and local variations	SPICE
AES	25.9/1000 (2.59%)	10.8/100 (10.8%)
Cortex-M0	12.8/1000 (1.28%)	5.1/100 (5.1%)
MIPS	20.2/1000 (2.02%)	1.7/100 (1.7%)
Total	59.9/3000 (1.96%)	17.6/300 (5.9%)

Delay Estimation with Calibration

We setup two experiments to evaluate our calibration method in Section 2.1.3. First, we shift both chip and DDRO supply voltages from nominal supply voltage ($0.9V$) to $0.8V$. This experiment setup represents the typical scenario where the nominal PVT corner is shifted. Second, we keep the chip supply voltage at $0.9V$ but shift all DDRO supply voltages to $\{0.8V$, $0.9V$, $1.0V\}$. This experiment setup captures the scenario where there is systematic within-die variation between the chip's critical paths and the DDROs (e.g., voltage drop in chip's power delivery network).

For each testcase, we simulate the critical paths (obtained from *MIPS*) and DDRO delays using Monte Carlo SPICE with 100 trials. Based on the simulation results, we estimate chip delay using the cluster-based method in Section 2.1.2 with five DDROs ($A_{user} = 50\%$) and compare it with the simulated chip delay. Among the 100 trials, we randomly choose a subset of the chip samples and apply the calibration procedure described in Section 2.1.3. Since the delay estimation is affected by the selection of chip samples, we repeat this experiment 50 times and report the average values of mean delay-estimation error.

Results in Table 2.6 show that when both chip and DDROs' voltages are at the nominal corner ($0.9V$) the mean delay-estimation error is only 1.25% without applying any calibration. Even when both chip and DDROs' voltages are shifted to 0.8V, the estimation error is only 1.70%. However, if chip voltage remains at $0.9V$ but DDRO voltage is shifted to $0.8V$ or $1.0V$, the estimation error increases significantly (12% to 21%). The estimation error can be reduced significantly when we apply our calibration method (Section 2.1.3). As the number of samples increases, the average mean delay estimation error reduces rapidly. For instance, the maximum of the average mean delay-estimation error is less than 2.5% with 30 samples.

Proof of Concept Silicon Results

We have taped out a testchip with DDRO-based performance monitoring using a foundry $45nm$ SOI technology with dual-V_{th} libraries. The testchip has an ARM *Cortex-M3* microprocessor [225] with DDROs. To synthesize the DDROs, we extract the critical paths from the microprocessor and cluster their sensitivities into five clusters by using the kmeans++ algorithm [7]. The results of the path sensitivities clustering is shown in Figure 2.3.[8] Then, for each cluster, we synthesize a DDRO which has delay sensitivities similar to the mean delay sensitivities of paths in the cluster. The synthesis method is the same as that in Section 2.1.3.

[8] At the time of our testchip tapeout, the clustering method for the problem in (2.8) had not yet been developed.

Table 2.6: Average of mean delay-estimation error normalized to mean chip delay. *MIPS* with 100 SPICE Monte Carlo trials.

Number of samples	Chip voltage = 0.8V	Chip voltage = 0.9V		
	DDRO voltage	DDRO voltage		
	0.8V	0.8V	0.9V	1.0V
1	6.82%	6.73%	5.78%	3.65%
2	5.21%	6.34%	5.42%	2.71%
5	3.06%	7.17%	2.51%	2.16%
10	3.21%	2.61%	2.00%	2.18%
15	2.56%	2.37%	1.83%	1.70%
20	2.05%	2.22%	1.46%	1.75%
25	1.99%	2.59%	1.41%	1.88%
30	2.37%	1.99%	1.45%	1.72%
35	1.88%	1.96%	1.40%	1.85%
50	1.77%	1.84%	1.39%	1.76%
100	1.74%	1.64%	1.22%	1.52%
No calibration	1.69%	20.68%	1.25%	12.34%

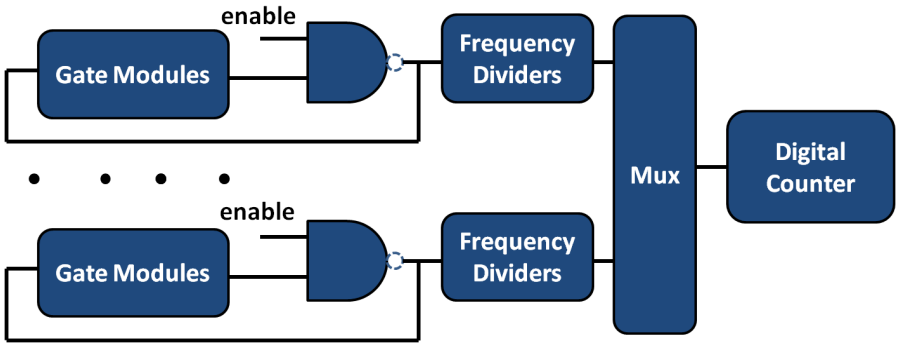


Figure 2.15: RO block schematic. In this testchip, we use a 12-stage frequency divider.

To control DDRO oscillation, a NAND (or AND) gate is added in each RO as shown in the schematic in Figure 2.15. An on-chip digital counter is used to obtain the RO frequencies, i.e., the counter will count the number of cycles of a RO within a measurement window. We repeat RO measurements with $40ms$ and $100ms$ measurement windows and measure the ROs in different sequences to make sure that the results are consistent and that systematic measurement error is minimized. For comparison, we have also implemented inverter-based ROs. The design information of the *Cortex-M3* and ROs is listed in Table 2.7.

Table 2.7: Design information of the testchip.

Component	Cell count	Cell type
Cortex-M3	50196	mixed VT
DDRO1	13+100	mixed VT
DDRO2	13+85	mixed VT
DDRO3	13+100	mixed VT
DDRO4	13+90	mixed VT
DDRO5	13+85	mixed VT
Inverter RO1	13+21	RVT
Inverter RO2	13+21	HVT
Inverter RO3	13+61	RVT
Inverter RO4	13+61	HVT
Inverter RO5	13+61	mixed VT

The RO cell count includes the additional NAND (or AND) gate and a 12-stage frequency divider (total 13 cells). The testchip layout and die photo are shown in Figure 2.16. We measured the processor maximum operating frequency and RO frequency using the testbed shown in Figure 2.17. There are two microcontroller units (MCUs) on the testbed. One of the MCUs is used to control the digital counter of the RO block and to measure the frequency of the ROs. The other MCU is used to control the processor and the on-chip PLL. We measure chip frequency by running a test program (fast Fourier transform) and increasing the processor's clock frequency (through PLL) until the processor generates incorrect results compared to the precalculated golden results. For each chip, we supply both RO and processor with the same supply voltage.

The measured mean chip and RO delays (14 testchips) are about two times of the corresponding simulation results. This suggests that the chips are operating at a very skewed PVT corner compared to the SPICE simulation. Therefore, we use the calibration method described in Section 2.1.3 to estimate chip delays. To minimize the estimation error we use all 14 chips for

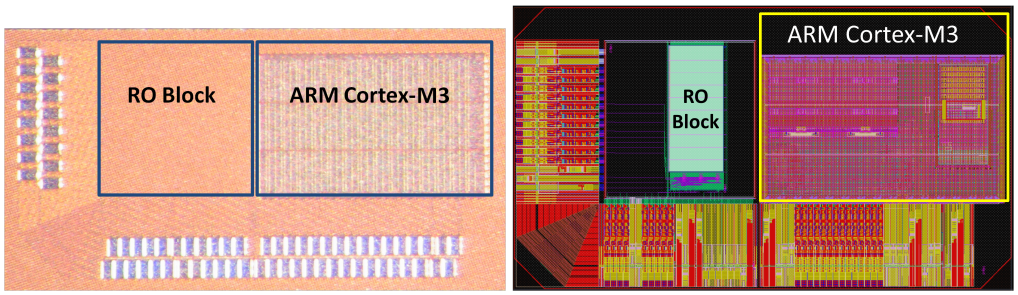


Figure 2.16: Testchip die photo and layout illustration.

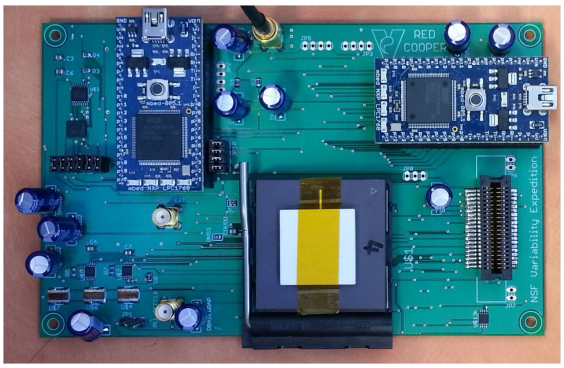


Figure 2.17: Testbed for RO frequency measurement and processor frequency measurement. Two microcontroller units are designed to control the processor and RO blocks, respectively.

the calibration. For each inverter-based RO, we treat it as one DDRO designed for the all critical paths, i.e., $x = y = 1$. Then we apply the same calibration as in Section 2.1.3 and estimation method as in Section 2.1.2 for the inverter-based ROs (with $a_{x,y} = 1$). The results of the mean delay-estimation error are shown in Figure 2.18 ($A_{user} = 0.5$). The measurement results show that by using five DDROs, we can reduce the mean delay-estimation error by 35% (from 2.3% to 1.5%) compared to generic inverter-based ROs. To ensure that our results are not sensitive to measurement errors, we repeat the analysis by injecting random noise (standard normal distribution with $\sigma = 1\%$, 3% and 5% with respect to RO frequency) into all RO measurements. Results in Table 2.8 show the average mean delay-estimation error of DDRO and inverter-based ROs across 30 random trials. The improvement of DDRO over inverter-based ROs is approximately 25% to 30%, which is consistent with our observation drawn from Figure 2.18.

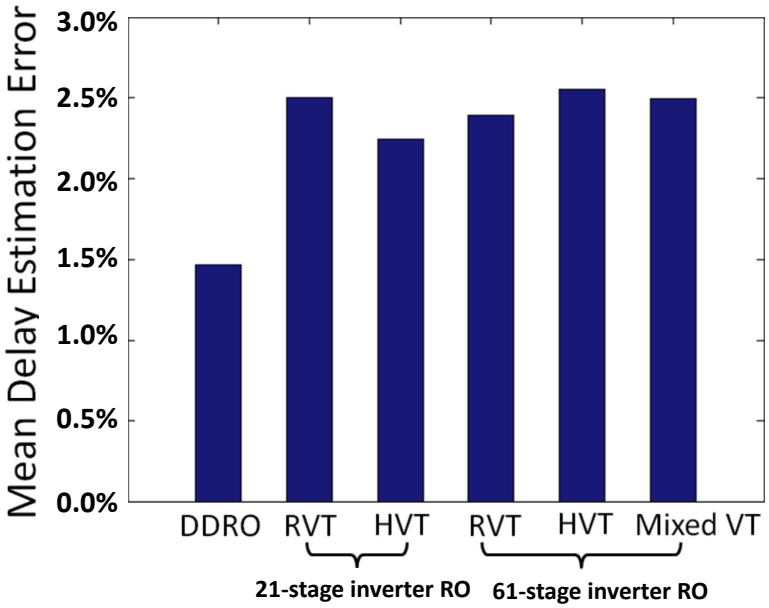


Figure 2.18: Mean delay-estimation error obtained from DDROs and inverter-based ROs. Estimation errors are calculated by taking the absolute difference between normalized estimation and normalized chip delay.

We also deploy ROs with different numbers of stages to estimate the effect of local variation. The results in Figure 2.18 show that the errors of 61-stage inverter ROs are similar to those of their 21-stage counterparts. This suggests that random local variation in ROs has little impact on the estimation error in our experiment. In Figure 2.19 we plot the statistics of the delay estimations. The results show that the minimum and maximum delay-estimation errors using DDROs are smaller compared to those obtained using the inverter-based ROs. Note

Table 2.8: Measurement error sensitivity analysis.

	Mean delay-estimation error					
	σ noise = 1%		σ noise = 3%		σ noise = 5%	
	Avg (%)	Improvement (%)	Avg (%)	Improvement (%)	Avg (%)	Improvement (%)
DDRO	1.60	NA	2.40	NA	3.30	NA
21-stage RVT inverter RO	2.60	38	3.20	25	4.40	25
21-stage HVT inverter RO	2.30	30	3.20	25	4.50	27
61-stage RVT inverter RO	2.50	36	3.20	25	4.40	25
61-stage HVT inverter RO	2.70	41	3.60	33	4.70	30
61-stage mixed VT inverter RO	2.60	38	3.40	29	4.40	25

that our results are based on measurements on 14 testchips from a single wafer. With multiple wafers from different lots, we expect that the improvements may be different (improvement is likely to be higher since the magnitude of global variation will increase compared to that of local variation).

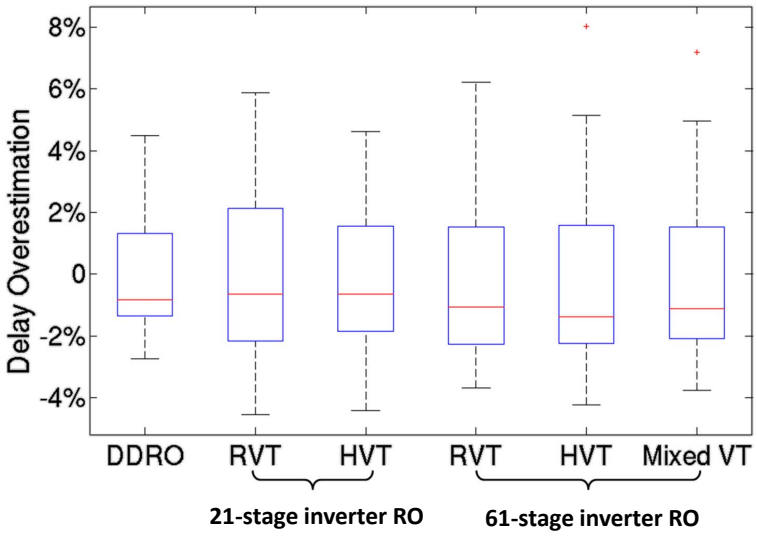


Figure 2.19: Maximum and minimum delay overestimation obtained from DDROs and inverter-based ROs. The edges of the boxes are the corresponding 25^{th} and 75^{th} percentiles of the data.

Comparison with Other Monitoring Methods

Table 2.9 summarizes the differences among different replica-like design-dependent monitoring methods. The method proposed in [137] has small implementation overheads be-

cause it uses a single representative critical path to estimate chip delay. Although this method does not require any calibration, it is relatively less accurate because it relies on a single representative critical path to estimate a set of critical paths.[9] The method of [36] also has small implementation overheads because it requires only a set of simple ROs. However, one-time calibrations at skewed process corners are required to make the ROs to be design-specific. Even with calibration, the method in [36] is not necessarily accurate because it calibrates the configurations of ROs to guardband for the worst possible delay. Tunable replica circuits in [65] are more accurate but require more complex circuits and calibration steps. By constrast, we propose a method which also has small implementation overheads because the monitor consists of only a few DDROs. Our method requires a calibration step to compensate for any difference between simulation model and actual silicon as described in Section 2.1.4. We expect that our method is more accurate than the method of [137] because we use multiple DDROs to track the delays of critical paths. Our method is also more accurate than [36] because we estimate the critical path delays instead of the worst-possible delay. Although our method may be less accurate than the tunable replica circuit, our method does not require calibration for every chip and also has less implementation overhead.

Table 2.9: Comparison of different replica-like design-dependent monitoring methods.

	Implementation overheads	Calibration effort	Accuracy
[137]	Small	No calibration	Low
[36]	Small	Low	Low
[65]	Medium	High	High
This work	Small	Low	Medium

2.1.5 Conclusions

We have proposed methods to systematically design multiple DDROs, and to estimate circuit performance (chip delay) based on the measurements from the multiple DDROs. Our study shows that our delay estimation method can achieve similar results as the path-based method with significantly less bookkeeping overhead. We also show that by using multiple DDROs we can reduce the mean delay overestimation by up to 25% (from 4% to 3%). The reduction is mainly limited by local variation, which cannot be captured by replica-like monitors. Further delay overestimation reduction will require in-situ type monitors, which have much higher area and design implementation overheads. We also observe that the benefit of using replica-like monitors (such as DDROs) is more significant when the local variation is relatively

[9] This approach is similar to our DDRO method (see Chapter 2.1.3) with $N_{ro} = 1$.

less compared to the global variation. If local variation dominates, then in-situ monitoring, although expensive, will fare better. With shrinking feature dimensions, increasing wafer sizes and changing device structures (e.g. fully depleted SOI, FinFETs), it is difficult to project which of the two components of variation is going to dominate in future technologies.

To verify the performance of DDROs and our delay estimation approach, we have taped out a testchip using foundry $45nm$ SOI technology together with an ARM *Cortex-M3* CPU. Our silicon results show that DDRO can reduce the mean delay-estimation error by 35% (from 2.3% to 1.5%) compared to generic inverter-based ROs.

2.2 Tunable Sensors for Process-Aware Voltage Scaling

Process variation is a critical aspect of VLSI circuit design because it causes wide performance spread [21] [117]. To recover excess margin allocated for process variation, many adaptive voltage scaling (AVS) techniques have been proposed [40] [69] [135] [148] [158].

AVS techniques can be classified as either open- or closed-loop. A typical *open-loop* AVS system utilizes a precharacterized lookup table (LUT) to find the corresponding minimum supply voltage for a given chip frequency target [40] [135]. Since the open-loop technique does not have a feedback mechanism, the LUT is heavily guardbanded to ensure reliable system operation. At the same time, characterizing the LUT is a time-consuming and expensive procedure, especially for a system-on-chip (SoC) design which has multiple operating modes and IPs.

A *closed-loop* AVS system adjusts supply voltage by probing actual chip performance, using on-chip monitors instead of using a LUT. To track timing performance of a chip, many critical path replica or in-situ monitor approaches have been proposed [60] [65] [69] [73] [137] [148] [174] [184]. However, the "critical paths" in a multiple-IP SoC design are not clearly defined, as chip performance depends on both operating modes and interactions among the IPs. Moreover, there are cases where exact input vectors to exercise worst-case timing paths in an SoC are not known during design time.

In this section, we propose an approach to design sensors for process-aware voltage scaling (PVS). Instead of designing performance monitors to track the timing performance of critical paths, we design ROs which have the worst-case *voltage scaling characteristics* across the entire process condition (see Section 2.2.1 for the details of voltage scaling characteristics). We design the PVS ROs such that they require a relatively higher supply voltage compared to critical paths of a SoC to compensate process variation-induced frequency drift. Therefore, any SoC manufactured in the process can safely perform a closed-loop AVS by using these ROs as hardware performance monitors. A new analysis of voltage scaling characteristics is a key enabler to our PVS methodology. Design ROs for worst-case voltage scaling characteristics is distinguished from a conventional RO-based monitoring method (e.g., [27]) which uses an arbitrary RO.

Application examples (scenarios) for the proposed ROs are shown in Figure 2.20. At the design stage, we design the PVS ROs using SPICE models and standard cells. Since there will be some difference between simulation and the silicon data, a silicon characterization step is required to calibrate the error between simulation and silicon data. At the silicon characterization stage, sample testchips at different process corners are provided by the foundry. In this stage, we

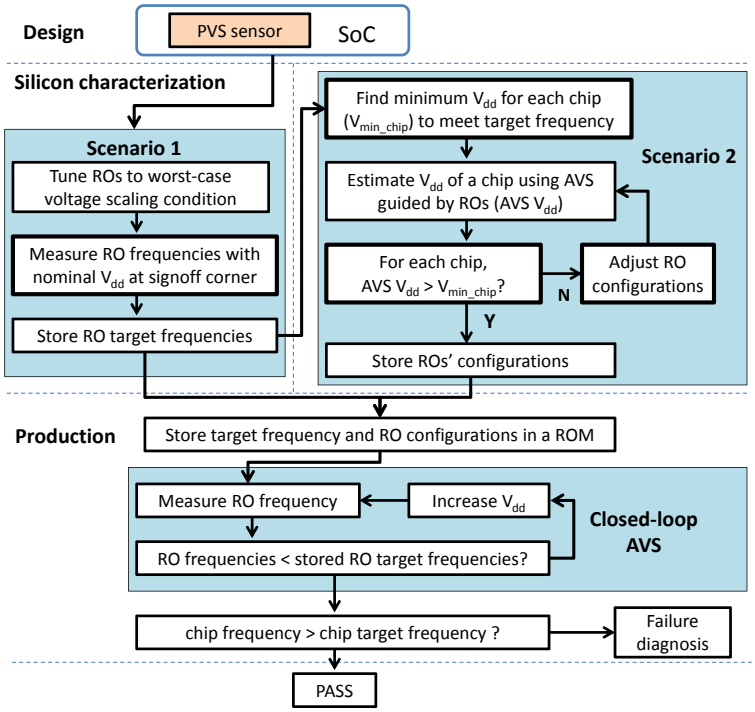


Figure 2.20: An application example for the proposed tunable ROs.

measure the ROs' frequencies with nominal operating voltage (V_0). The frequencies measured at the signoff corner (e.g., SS corner) will be used as the target frequencies of the ROs during AVS (Scenario 1). In this application scenario, our ROs have no information about the design, and they are designed to guardband for the worst-case voltage scaling characteristics. Therefore, the AVS guided by our ROs will always overestimate the supply voltage needed for a chip to meet its operating frequency. The excess supply voltage can be reduced when chip maximum frequency f_{max} is also measured during the silicon characterization stage (Scenario 2). In this scenario, we can tune the voltage scaling characteristics of the ROs such that for each chip in the silicon characterization stage, the supply voltage suggested by the AVS (guided by the ROs) is slightly higher than the minimum voltage (V_{min_chip}) needed for a chip to meet its required operating frequency. When all test chips manufactured for silicon characterization can safely operate at their respective operating frequencies using AVS guided by the PVS ROs, we record the configurations of the ROs. In this characterization step (Scenario 2), the testchips are manufactured at biased process corners. Thus, calibrating the ROs with these testchips will configure the ROs to account for circuit performance variation due to widely spread process variation. Sampling the testchip at different process corners is important because this allows the

configurations of the ROs to be applied in the subsequent production stage without additional calibrations.

To capture the within-die systematic process variation, we can place multiple copies of the ROs in a chip (e.g., a set of ROs for every $1mm^2$ area on the chip). However, the effect of within-die random variation cannot be captured by our method due to the nature of the replica-type monitoring approach. Thus, additional timing or voltage margin must be added to ensure reliable circuit operation. Meanwhile, by having multiple copies of the ROs in a chip, the effect of within-die temperature variation on circuit performance can be also captured by the ROs.

During mass production, the previously obtained ROs' configurations will be stored in every production chip. Then, we run AVS tests with the stored ROs' configurations and RO target frequencies. If a chip fails to meet its target frequency with the AVS guided by PVS ROs, this means that either the calibration during silicon characterization is inaccurate or the chip has failed due to other reasons. After studying the root cause of the failure, the silicon characterization step can be modified if necessary (e.g., adjust ROs' configurations such that the AVS is less aggressive in reducing supply voltage).

Note that in Scenario 1, we skip the procedures of Scenario 2, and all ROs are configured to the worst-case voltage scaling condition. Although this approach leads to a more pessimistic AVS, the tunability of the ROs allows the chip customer to recover the pessimism in AVS by calibrating RO configurations. Since the PVS ROs are design-independent, a PVS IP can be embedded in different SoCs to support AVS. For example, PVS ROs can be deployed within a performance monitor block in a power management IP such as [257].

Our method is different from critical path-driven tunable circuits [65] [69]. First, critical path replica techniques design the replica to be flexible to match the timing performance of a set of critical paths. Because of the inherent design intention to match the timing performance, the design of a critical path replica is dependent on the circuit to be matched (e.g., the TRC must have the flexibility to match the total critical delays). By contrast, we design our tunable ROs such that they can be configured to have different voltage scaling characteristics. This difference in design intention is important because, as we will show, matching the voltage scaling characteristics of different circuits can be achieved by having a set of tunable ROs which are design-independent. As a result, we can optimize the ROs and reuse them in other designs. Second, our proposed method only calibrates the ROs at the silicon characterization stage. After this calibration step, the settings will be applied to all production chips instead of calibrating the ROs for every production chip. Since per-chip calibration is not required, our method saves

testing time during chip production. We summarize our contributions as follows.

- We propose a simplified process-aware voltage scaling methodology and analyses of the worst-case condition of voltage scaling under process variation.
- We propose circuit techniques to tune the voltage scaling characteristic of the sensor such that it has flexibility to mimic the voltage scaling characteristics of a chip across a range of process variations. With the tunability, we can reduce the supply voltage by up to $30mV$ (compared to non-tunable ROs) without causing any timing violation.
- Our tunable sensor is design-independent, and can therefore be embedded in any other IPs.

2.2.1 Process-Aware Voltage Scaling

Overview of PVS

Figure 2.21 shows the basic idea of the PVS methodology, wherein we model the frequency of a critical path as a linear function of supply voltage (V).[10] We denote the frequency of a critical path by $f_{path}(j, k, V)$ where j is the index of a critical path, k denotes the process condition, and V is the supply voltage. Similarly, we define the frequency of an RO by $f_{ro}(y, k, V)$, where y is the index of a RO.

We define the target frequency of the critical paths f_{tar_path} as the minimum frequency of all critical paths at nominal voltage V_0. Note that the target frequency is specific to the signoff corner. Unless otherwise specified, we define the target frequency at the SS corner, i.e.,

$$f^{ss}_{tar\_path} = \min_{j=1}^{N_{path}} f_{path}(j, SS, V_0)$$

where N_{path} is the total number of critical paths, V_0 is the nominal voltage and $f^{ss}_{tar\_path}$ is the target frequency of the chip at the SS signoff corner.

When a circuit is manufactured at process condition k (dashed line in Figure 2.21), the frequency of the circuit is significantly higher than $f^{ss}_{tar\_path}$. Thus, we can perform voltage scaling to reduce the power of the circuit as long as the circuit meets the targeted frequency. The minimum voltage required for a critical path j to meet its targeted frequency at a process condition k is denoted as $V_{min\_path}(j, k)$. When there is more than one critical path, the minimum

[10] This approximation simplifies calculation while introducing small error [69].

voltage for a circuit $V_{min\_chip}(k)$ is given by

$$V_{min\_chip}(k) = \max_{j=1}^{N_{path}} V_{min\_path}(j,k) \tag{2.19}$$

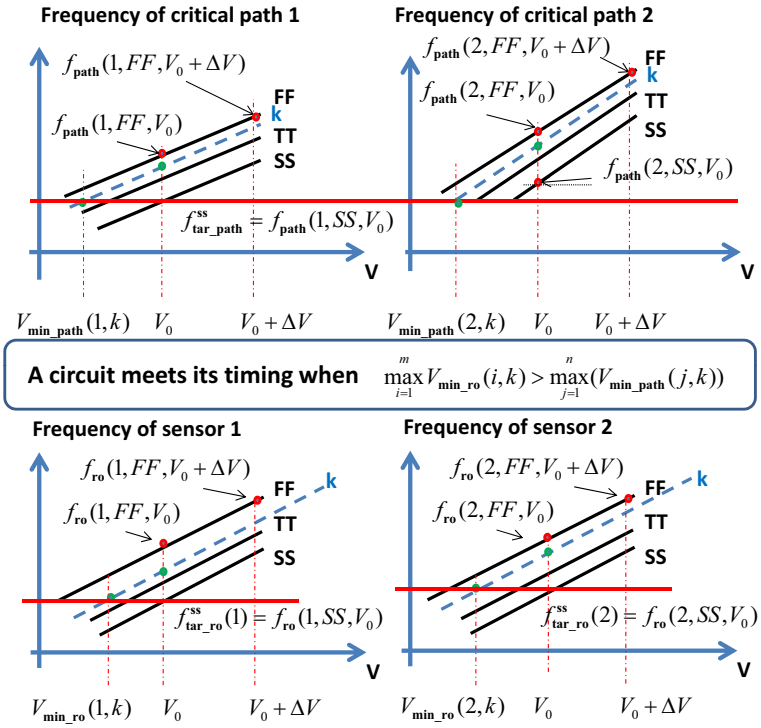


Figure 2.21: Illustration of process-aware voltage scaling.

As mentioned above, finding the exact critical paths in an SoC to calculate $V_{min\_chip}(k)$ is very difficult. Therefore, we propose to adjust the supply voltage of a circuit by measuring the frequencies of on-chip ROs. As shown in the lower part of Figure 2.21, the frequency of the y^{th} RO is represented as $f_{ro}(y,k,V)$. The target frequency of each on-chip RO ($f^{ss}_{tar\_ro}(y)$) is defined at the same signoff corner as the circuit, e.g., $f^{ss}_{tar\_ro}(y) = f_{ro}(y,SS,V_0)$, and each RO has a specific target frequency. We denote $V_{min\_ro}(y,k)$ as the minimum voltage for the y^{th} RO to meet its targeted frequency, where k represents the process condition of the RO. By measuring the RO frequencies at two or more supply voltages, we can extract each RO's frequency versus voltage "slope", and calculate $V_{min\_ro}(y,k)$ from the equation

$$V_{min\_ro}(y,k) = V_0 - \frac{(f_{ro}(y,k,V_0) - f^{ss}_{tar\_ro}(y))\Delta V}{f_{ro}(y,k,V_0+\Delta V) - f_{ro}(y,k,V_0)} \tag{2.20}$$

where ΔV is the difference between the nominal voltage and chip's supply voltage during RO

measurement. After obtaining $V_{min\_ro}(y,k)$, we can use it as a reference to scale the supply voltage of the chip. A chip will still meet its performance target as long as $V_{min\_ro}(y,k)$ is larger than $V_{min\_chip}(k)$. Thus, the "safe voltage scaling condition" for a chip is defined as

$$V_{min\_chip}(k) < \max_{y=1}^{N_{ro}}\{V_{min\_ro}(y,k)\}, \forall\, k \tag{2.21}$$

where N_{ro} is the total number of ROs. To ensure that the chip meets its targeted frequency, we scale the supply voltage of the chip to

$$V_{min\_est}(k) = \max_{y=1}^{N_{ro}}\{V_{min\_ro}(y,k)\} \tag{2.22}$$

Fundamental Properties of PVS

Equation (2.20) shows that the minimum scaling voltage of a RO (or a critical path) is determined by two fundamental properties:

1. **Process distance**: $f_{ro}(y,k,V_0) - f^{ss}_{tar\_ro}(y)$
2. **Scaling rate** : $(f_{ro}(y,k,V_0+\Delta V) - f_{ro}(y,k,V_0))/\Delta V$

Process distance is the process-induced frequency shift relative to target frequency. This property is usually modeled as a random variable due to the randomness in manufacturing processes. However, it is also affected by the design of the circuit. For example, different critical paths have different sensitivities to sources of process variation. Another fundamental aspect of PVS is its formulation based on a *scaling rate* of frequency with respect to supply voltage. Clearly, this is also a circuit-related property which varies depending on the process condition.

Note that voltage scaling for a circuit is defined by relative value of the process distance and the scaling rate (i.e., process distance/scaling rate). Based on these properties, we can derive the voltage scaling characteristic of an arbitrary circuit. We are interested in studying the following questions.

1. Given a process technology, what is the range of voltage scaling defined by process distance and scaling rate?
2. What circuit techniques can be used to design a monitoring circuit with tunable voltage scaling characteristics?

Answering the first question helps to identify the worst-case voltage scaling condition, which is the design goal of our PVS ROs. Answering the second question gives us feasible design options to design PVS ROs to achieve the goal.

2.2.2 Circuit Analysis

Voltage Scaling Sensitivity

As mentioned above, the voltage scaling characteristic of a critical path is given by

$$\text{voltage scaling} \equiv \frac{\text{process distance}}{\text{scaling rate}} \equiv \frac{f_{path}(y,k,V_0) - f^{ss}_{tar\_path}(y)}{f_{path}(y,k,V_0+\Delta V) - f_{path}(y,k,V_0)} \tag{2.23}$$

To gain intuition about the sensitivity of voltage scaling to circuit parameters, we model $f_{path}(.)$ using the Elmore delay model [70].

$$\begin{aligned} f_{path}(y,k,V_0) &= \frac{2}{d_{nmos}(y,k,V_0) + d_{pmos}(y,k,V_0)} \\ d_{nmos}(y,k,V_0) &= \frac{R_{nmos}(k,V)}{W_{nmos}}(1+\beta)\cdot[W_{nmos}(\beta+1)C_{gate}(k)N_{fanout} + L*C_{wire}] \\ &\quad + L^2R_{wire}C_{wire} + LR_{wire}(\beta+1)C_{gate}(k)N_{fanout} \\ d_{pmos}(y,k,V_0) &= \frac{R_{pmos}(k,V)}{W_{nmos}\beta}(1+\beta)[W_{nmos}(\beta+1)C_{gate}(k)N_{fanout} + L*C_{wire}] \\ &\quad + L^2R_{wire}C_{wire} + LR_{wire}(\beta+1)C_{gate}(k)N_{fanout} \end{aligned} \tag{2.24}$$

where L is wire length, W_{nmos} is channel width of NMOS, N_{fanout} is the fanout of the driver, R_{wire} is wire resistance per μm, C_{wire} is wire capacitance per μm, β is the beta ratio between PMOS and NMOS channel width, $C_{gate}(k)$ is gate capacitance per μm channel width, and $R_{nmos}(k,V)$ and $R_{pmos}(k,V)$ are effective drive resistance of NMOS and PMOS, respectively. To study the sensitivity of voltage scaling, we extract parameters in Equation (2.24) from an inverter of a $65nm$ foundry library. The values of $R_{nmos}(k,V)$ and $R_{pmos}(k,V)$ are calculated by using effective current approximation [6],

$$\begin{aligned} R_{\{nmos,pmos\}}(k,V) &= \frac{2V}{I_L + I_H} \\ I_L &= I_{ds} \text{ when } V_{gs} = V/2, V_{ds} = V \\ I_H &= I_{ds} \text{ when } V_{ds} = V/2, V_{gs} = V \end{aligned}$$

where I_L and I_H are the drive currents (I_{ds}) of a MOS transistor at different bias conditions. The parameters and effective currents are summarized in Table 2.10.

Table 2.10: Technology parameters of a $65nm$ library.

Parameters	Process corners		
	SS	TT	FF
W_{nmos} (μm)	0.09	0.09	0.09
R_{wire} ($\Omega/\mu m$)	0.16	0.16	0.16
C_{wire} ($fF/\mu m$)	0.00017	0.00017	0.00017
C_{gate} ($fF/\mu m$)	1.03	1.09	1.16
I_L NMOS, 1.0V (μA)	52	134	258
I_L NMOS, 0.9V (μA)	29	87	192
I_H NMOS, 1.0V (μA)	459	591	723
I_H NMOS, 0.9V (μA)	348	470	594
I_L PMOS, 1.0V (μA)	29	66	125
I_L PMOS, 0.9V (μA)	16	41	88
I_H PMOS, 1.0V (μA)	232	294	353
I_H PMOS, 0.9V (μA)	172	227	281

Using the parameters in Table 2.10, from Equations (2.23) and (2.24) we calculate V_{min} of the inverter for TT corner (i.e., $k = TT$) and its sensitivities. First, we calculate the nominal V_{min} of the inverter with L = 10μm, W_{nmos} = 1μm, β = 1.5, N = 1. Then, we sweep the value of the L, W_{nmos}, β, N, R_{nmos} and R_{pmos} parameters, one at a time (other parameters remain at their nominal values), from 0.2 to 4 times of their nominal values, to evaluate the effect of each parameter on V_{min}. The results in Figure 2.22 show that V_{min} is most sensitive to R_{nmos} and R_{pmos}, followed by β, L, fanout, and W_{nmos}. We also observe that when the value of each parameter is increased, its impact on the value of V_{min} becomes smaller. V_{min} changes rapidly as the (normalized) parameter values scale below 1.0. There are also practical lower limits for the parameters. For example, the driver size (W_{nmos}), fanout, R_{nmos}, etc. cannot scale down to zero. Hence, voltage scaling of a circuit has finite bounds. From our studies, we also observe that V_{min} can be significantly lower (resp. higher) when we only consider d_{nmos} (resp. d_{pmos}) in Equation (2.24).

Voltage Scaling Analysis Using SPICE Simulation

Although the previous analysis provides useful information regarding the sensitivities of V_{min} to circuit parameters, many effects are not captured by the simplified equations. To

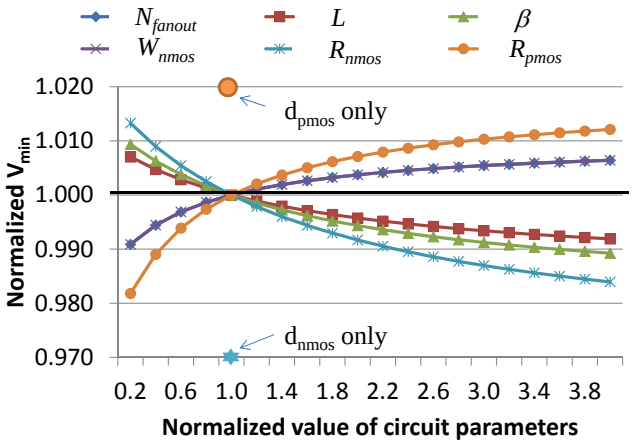


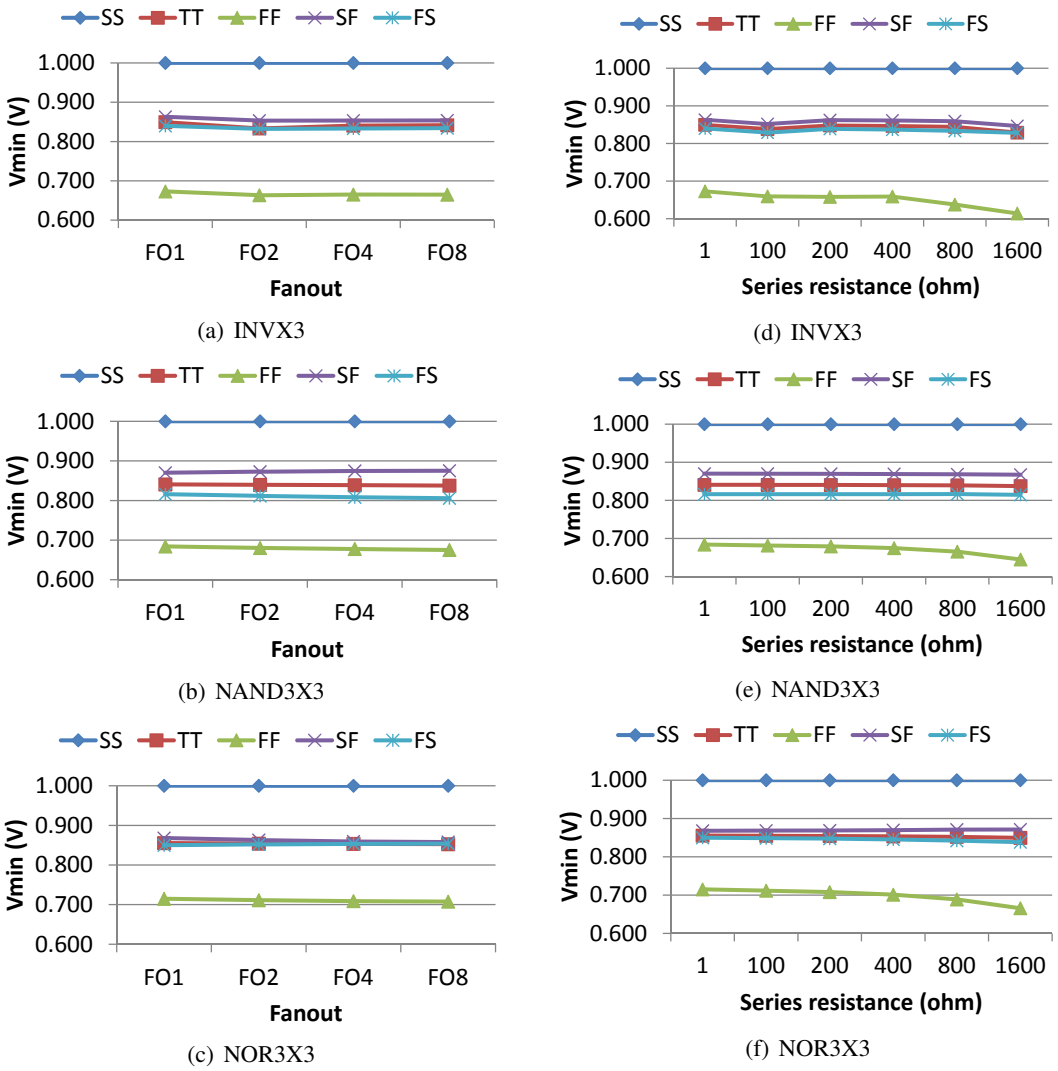
Figure 2.22: Sensitivity of V_{min} to circuit parameters.

investigate the range of voltage scaling as well as the effect of circuit parameters, we simulate different ROs with different configurations.

First, we evaluate the effect of fanout by adding dummy gates in every stage in the RO. Figure 2.23 shows that V_{min} extracted from the ROs is not sensitive to fanout for ROs implemented with different standard cells. Second, we increase the series resistance along the signal transition path of the ROs with fanout = 1. Figure 2.23 shows that series resistance can affect V_{min} when the resistance value is large. For $65nm$ technology, the wire resistance per μm is approximately 0.16Ω. Therefore, V_{min} at 400Ω corresponds to the case where a $2.5mm$ wire is connected to the output of a driver. Since reasonable design usually does not permit such a long wire, it is safe to assume that wire resistance will not affect V_{min}. This implies that the voltage scaling characteristic of a chip is not affected by wire parasitics.

Third, we add passgates at the output of each driver of the ROs to study their effects on V_{min}. To study different scenarios, we also change the effect of the passgates by adding more passgates in parallel or in series. Results in Figure 2.24 show that adding passgates in parallel can change the V_{min} significantly. V_{min} increases when the number of parallel passgates is increased. This is because more passgates in parallel reduces the series resistance of the ROs. This result agrees with the estimations obtained in Equation (2.24), in which increasing L reduces V_{min}. Figure 2.24 shows that V_{min} changes only slightly when the number of series passgates is increased. This is because the effect of adding series resistances saturates as the sum of series resistance increases.

Equation (2.24) shows that R_{nmos} or R_{pmos} has significant impact on V_{min}. To study this, we simulate ROs with different standard-cell types. Results in Figure 2.25 show that V_{min}



(a) INVX3 (d) INVX3

(b) NAND3X3 (e) NAND3X3

(c) NOR3X3 (f) NOR3X3

Figure 2.23: SPICE simulations of ROs implemented with INV, NAND and NOR standard cells. The results show that V_{min} is not sensitive to the fanout and series resistance (except for large resistance values).

varies over ROs with different cell types. For example, we see that V_{min} of NOR-based ROs is larger than that of INV-based ROs. This is because the NOR-type standard cell has a stacked pull-up network with a larger R_{pmos} compared to the balanced pull-up and pull-down networks of an inverter. On the other hand, V_{min} of NAND-based ROs is smaller than that of INV-based ROs especially at TT and FS process corners. This agrees with the estimations obtained from Equation (2.24), where V_{min} is smaller for a larger R_{nmos} (a NAND gate has a larger R_{nmos} compared to an INV gate). However, the trend is not obvious at FF process corner. This may be due to layout parasitics and other second-order effects which are not modeled in our analysis.

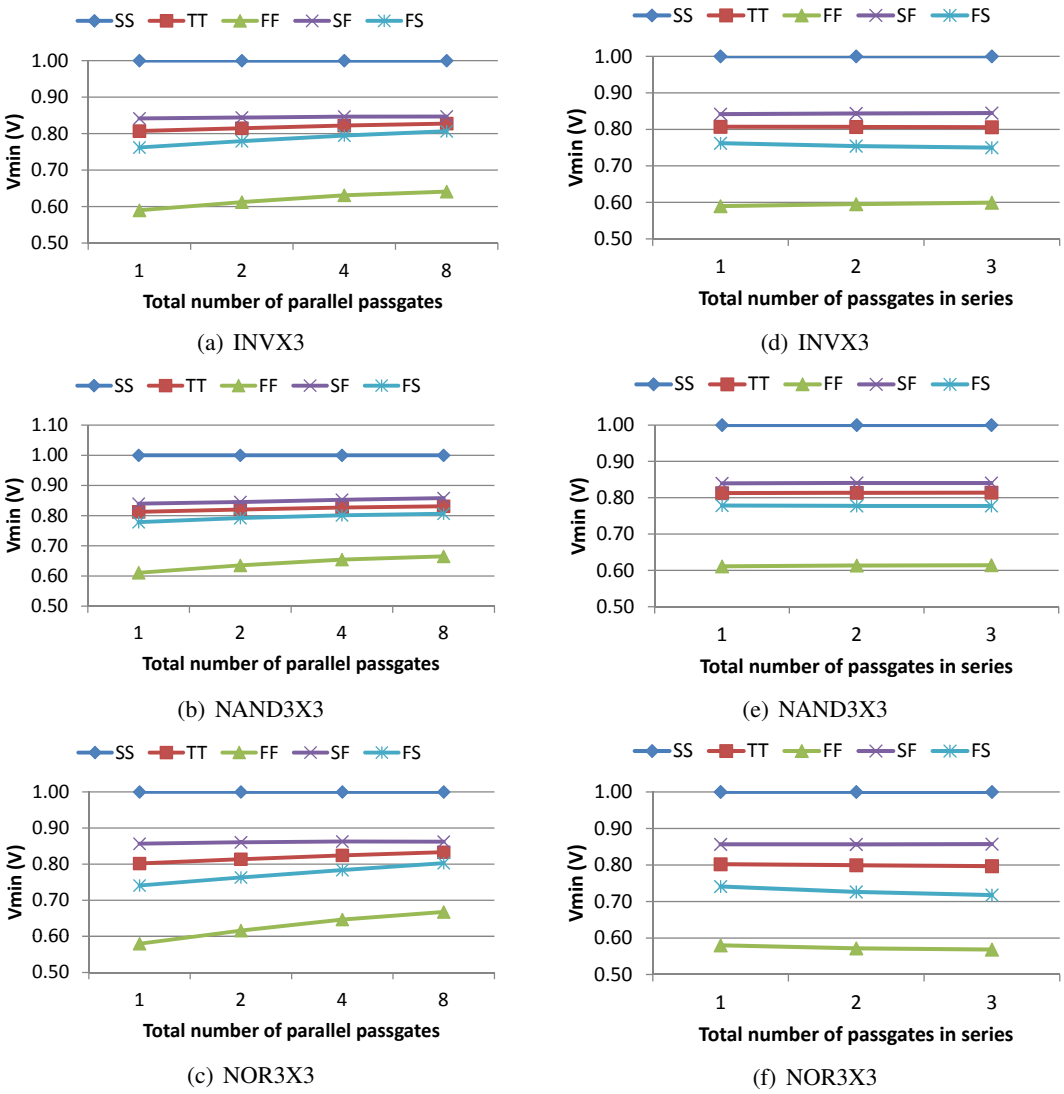


Figure 2.24: V_{min} increases when the number of passgates in parallel is increased. Adding more passgates in series has little effect on V_{min}.

Note that V_{min} increases sharply when the driver is increased from the minimum size (X0) to larger sizes. This is due to the diffusion height of the minimum-sized cell being significantly less than the row height of the standard cell. Thus, the layout parasitics of cells with the minimum driver size are typically different from those of other cells. Note that the maximum value of V_{min} at different corners is determined by the V_{min} of different cell types. For example, the NAND-based RO has the largest V_{min} at SF corner while the NOR-based RO has the largest V_{min} at FS corner. Therefore, we require ROs implemented with different cell types to ensure that we capture the worst-case scenario in voltage scaling.

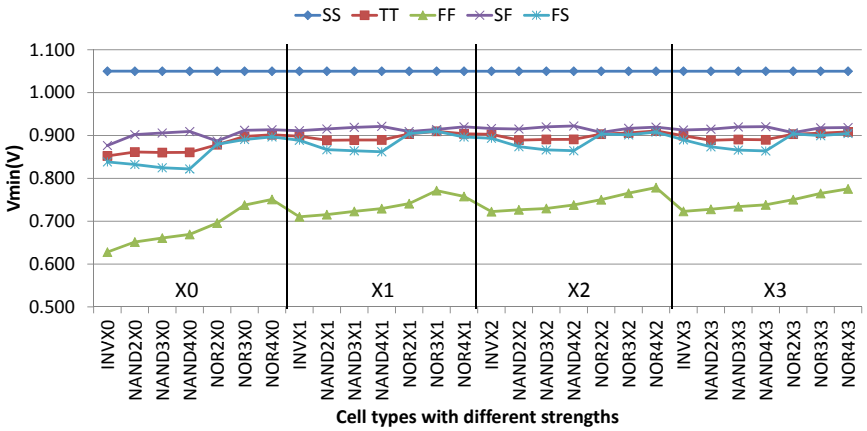


Figure 2.25: V_{min} varies across different cell types {INV, NAND2, NAND3, NAND4, NOR2, NOR3, NOR4} and strengths {X0, X1, X2, X3}.

2.2.3 Design of a Sensor with Tunable Voltage Scaling Characteristics

From the studies in the previous subsection, we observe that the voltage scaling characteristic of a circuit (RO) is mainly affected by the cell type. Among the circuit parameters, we only see significant changes in V_{min} when we add passgates in parallel to the ROs. Thus, we design our PVS sensor with different cell types and use passgates in parallel to tune the characteristic of the ROs. Our PVS sensor design seeks to achieve two main goals:

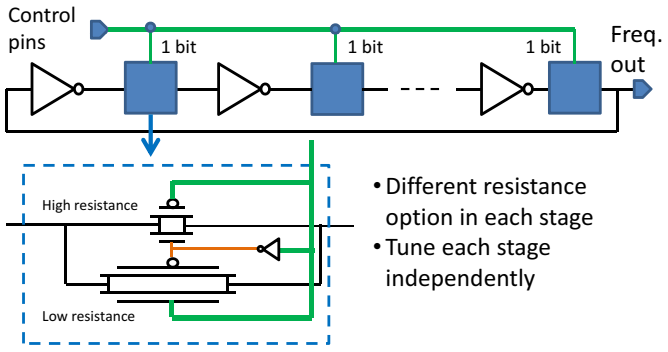
1. maximize the range of V_{min}; and
2. ensure that tunability of the sensor (V_{min} versus RO configuration) is consistent across different process corners.

Here, we present two of the circuit approaches that we have investigated to achieve these goals. The circuits are illustrated in Figure 2.26.

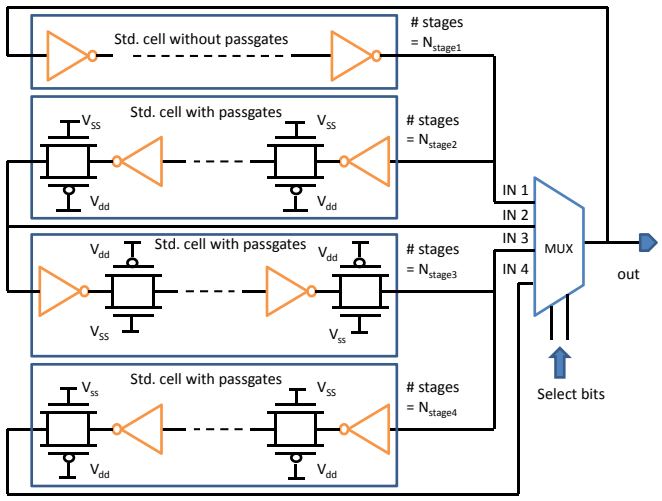
In the first approach, we add a pair of passgates in parallel at every stage of a RO, one with minimum-sized devices and the other with large device sizes. In this design, we can choose to turn on one passgate through a control pin assigned to the passgate. When we choose to turn on the passgate with minimum-sized devices, the high resistance passgate will reduce V_{min} and vice-versa when we turn on the passgate with larger device sizes. Although we can assign a control pin for each stage of the RO to achieve fine granularity, having a large number of control pins will incur higher design and area overheads. Since the voltage levels in an AVS system are discrete with coarse granularity, there is no need to have very fine granularity for the sensor. We divide the 33 stages of the RO into nine subsections (the last subsection has five stages whereas

all other subsections have four stages), with all passgates in each subsection sharing a control pin. Thus, only nine control pins are required instead of 33.

In the second approach, we divide an RO into several subsections with different number of stages (N_{stage1}, N_{stage2}, ...) and connect the output of the subsections to a MUX such that we can choose which subsection is included in the oscillation. For example, when we set the MUX select bits to $\{0, 0\}$, the output of the MUX is connected to "IN 1". As a result, only the first subsection is included in the oscillation. If we change the select bits to $\{0, 1\}$, then the first and second subsections are included. The advantage of this method is that through the MUX and select bits, we can bypass the cells with passgates, and achieve the maximum V_{min} of the RO



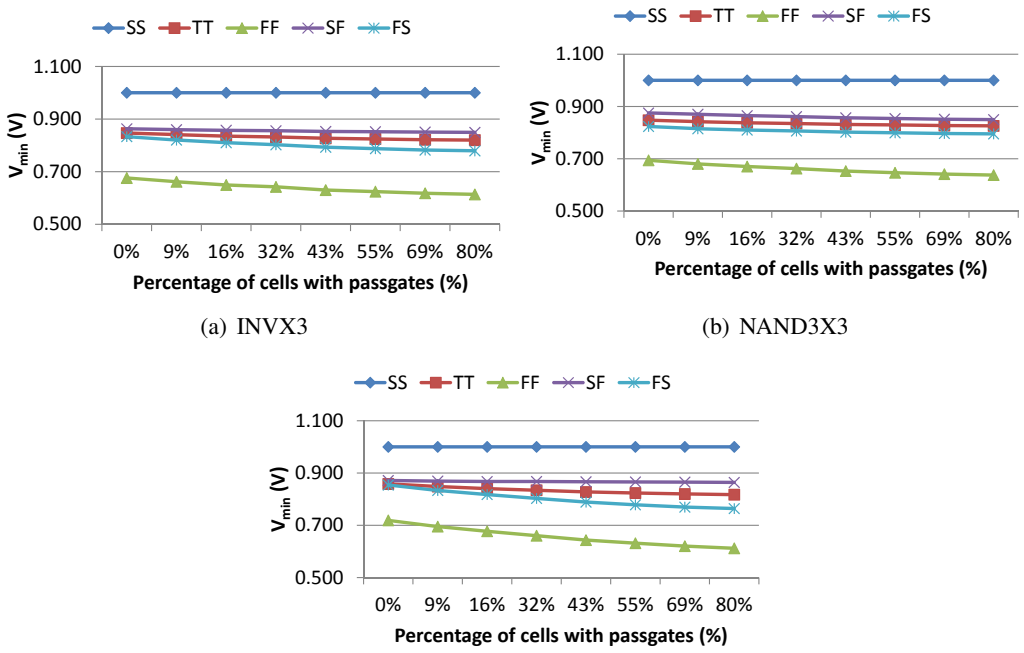
(a) We can use a MUX-like structure to control the ratio between different gates. Since V_{min} varies from one gate to another, we can connect different gates in series to achieve tunability of V_{min}.



(b) By controlling the select bits, we can change the number of series transistors along the signal transition path of the RO. This changes the effective resistance when the RO charges or discharges a node. As a result, this changes the V_{min} of the sensor.

Figure 2.26: Proposed tunable circuits.

(adding passgates will reduce V_{min}). Since the V_{min} of the RO is determined by the ratio of cells with passgates to cells without passgates, always including the first subsection could limit the tunability. For example, we need a large number of stages with passgates (and area) to increase the ratio of cells with passgates to cells without passgate.



(a) INVX3

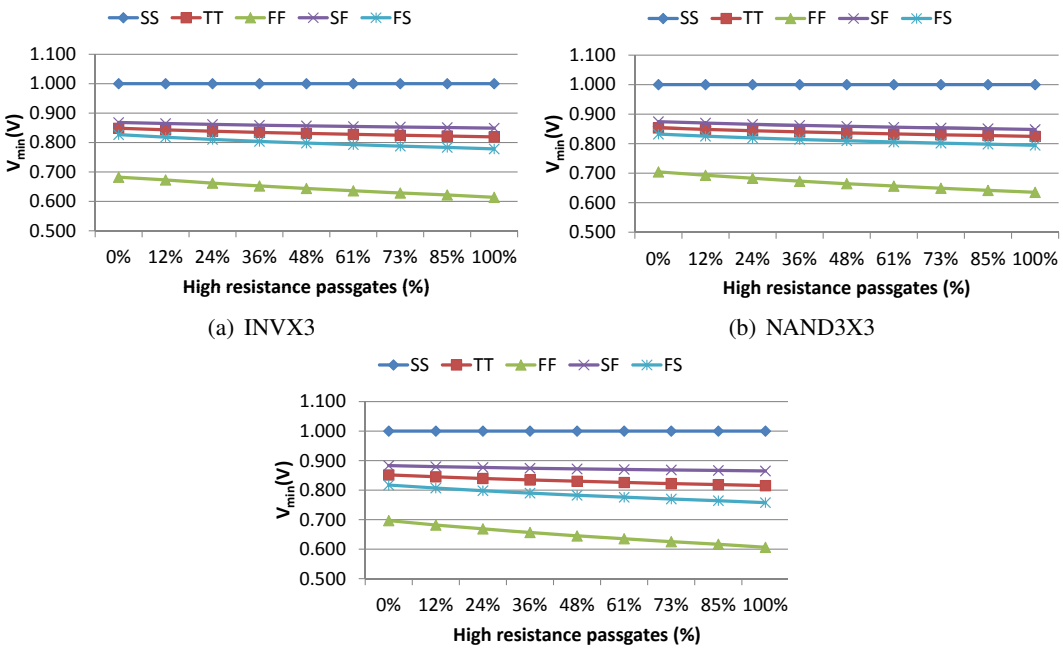
(b) NAND3X3

(c) NOR3X3

Figure 2.27: V_{min} is minimum when the RO consists of standard cells with passgates. By controlling the values of N_{stage1}, N_{stage2}, etc., we can control the percentage of cells with passgates, and achieve a linear relationship between V_{min} and the decimal values represented by the select bits of the MUX.

Simulation results in Figure 2.27 and Figure 2.28 show that both of these circuit approaches achieve similar ranges of tunability. Since the first approach has lower area overhead, we choose it for use in our simulation experiments. Based on the analysis in Figure 2.25, we observe that the maximum V_{min} is determined by different gate types, depending on the process conditions. To ensure that the ROs can have the maximum V_{min} across different process conditions, we choose to build the RO in Figure 2.26(b) with INVX3, NAND3X3 and NOR3X3 instances.[11] As mentioned above, the circuit option in Figure 2.26(b) has a slightly lower V_{min_ro} due to the passgates in the ROs. To ensure that V_{min_ro} of the ROs includes the worst-case voltage scaling characteristic, we add an additional $5mV$ margin to the V_{min_ro} in our simulation experiments.

[11] For gates with multiple inputs, we connect the inputs as a single net.



(a) INVX3

(b) NAND3X3

(c) NOR3X3

Figure 2.28: V_{min} of the proposed circuit for different standard cells. By controlling the percentage of cells with higher resistance, we can tune the V_{min} of the RO.

2.2.4 Experimental Results

In our experiments, we use three modules of the *OpenSPARC T1* processor [242] (Table 2.11). Module designs are implemented with a $65nm$ foundry library. The netlists are synthesized with *Synopsys Design Compiler* [250]. We extract critical paths of the modules in Table 2.11 at SS, TT and FF corners with *Synopsys PrimeTime vC-2009.06-SP2* [254]. For each process corner, we extract the top 100 critical paths and their corresponding SPICE netlists. We then simulate all the critical paths with *Synopsys HSPICE* [251] at SS corner, $V_0 = 1.0V$ and 125^oC to obtain the $f^{ss}_{tar\_path}$ of each module. The $f^{ss}_{tar\_path}$, power and area values of the implemented modules are given in Table 2.11.

Table 2.11: OpenSPARC T1 modules ($V_0 = 1.0V$).

	power (mW)	area (mm^2)	$f^{ss}_{tar\_path}$ (MHz)
SPARC_FPU	4.13	0.015	710.2
SPARC_TLU	438	0.098	506.6
SPARC_MUL	19.8	0.050	1042.1

Guardband Voltage Scaling

We perform an experiment to validate that our PVS sensors satisfy the "safe condition" in Equation (2.21) when the ROs are configured to have maximum V_{min_ro} (i.e., all passgates in the ROs have low resistance). To emulate process variation, we model threshold voltage of NMOS (V_{thn}) and PMOS (V_{thp}), channel length and oxide thickness as independent Gaussian random variables. The 3σ values of these variation sources are extracted from the foundry device model.[12] The mean (μ) and standard variation (σ) of the random variables are summarized in Table 2.12.

Table 2.12: Global variation parameters.

Variation source	μ	3σ
ΔV_{thn}	0	$30mV$
ΔV_{thp}	0	$30mV$
Δchannel length	0	$5.00nm$
Δoxide thickness	0	$0.06nm$

To estimate timing performance of the critical paths and ROs under process variations, we sample the variation sources randomly. We then apply the variations when running an SPICE simulation, and repeat this 100 times. This Monte Carlo experiment only includes global variation because our simulation setup does not support a local variation model.

Based on the simulated critical paths and RO delays, we calculate V_{min_chip} and V_{min_est} based on their definitions in Equations (2.19) and (2.22). Since there are INV-, NAND- and NOR-based ROs, V_{min_est} is the maximum V_{min_ro} of the three ROs. For comparison, we also include the results of non-tunable INVX3-, NAND3X3- and NOR3X3-chained ROs. These ROs are similar to our ROs, but there is no passgate in between consecutive stages.

Figure 2.29 shows that the voltage difference between V_{min_est} and V_{min_chip} is always positive. This implies that the sensors can be used to guardband the modules without calibration.

Optimizing Target Frequency for Margin Reduction

Our next experiment considers a scenario where V_{min_chip} of every chip is available to calibrate the PVS sensors. Hence, we can optimize the configuration (control bits) of the tunable

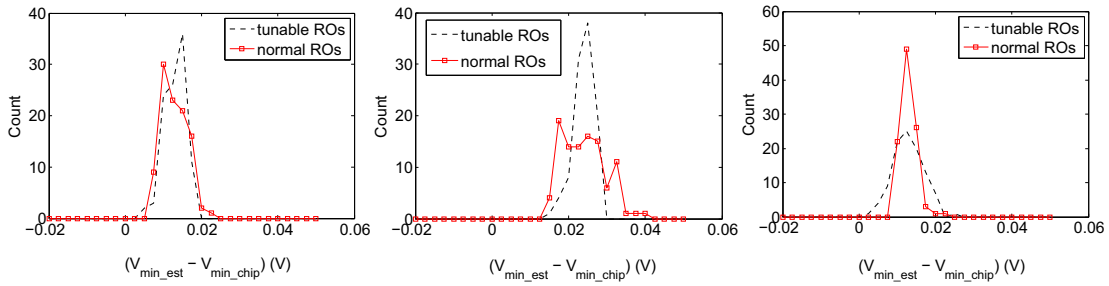
[12] We assume that process parameters at SS and FF corners define the $\mu \pm 3\sigma$ of the variation sources.

ROs to reduce supply voltage. The problem can be formulated as follows.

$$\begin{aligned} \text{minimize } & \sum_{k}\{V_{min\_est}(k) - V_{min\_chip}(k)\} \\ \text{subject to } & V_{min\_est}(k) > V_{min\_chip}(k), \forall\, k, y \\ & \max_{y}[V_{min\_ro}(y,k)|_{\psi(y)}] = V_{min\_est}(k) \\ & V_{min\_ro}(y,k)|_{\psi(y)} = V_0 + (f_{tar\_ro}(y)|_{\psi(y)} - f_{ro}(y,k,V_0)|_{\psi(y)}) \cdot \frac{1}{\alpha(y)|_{\psi(y)}} \end{aligned} \tag{2.25}$$

where $\alpha(y)$ and $\psi(y)$ respectively denote the scaling rate and configuration of the y^{th} RO. Note that $f_{tar\_ro}(y)$, $f_{ro}(y,k,V_0)$ and $\alpha(y)$ are all specific to $\psi(y)$. This ensures that V_{min_est} guided by our ROs is always less than V_0. This property is a key reason why the tunability in our circuit is different from using f_{tar_ro} as a means to adjust voltage scaling. For example, increasing f_{tar_ro} will cause the chip at SS corner to operate at a voltage higher than V_0, which may cause reliability-related failures. Since each INV-, NAND- or NOR-based RO has nine configurations, we calculate $V_{min\_est}(k)$ for all 729 combinations. After that, we compare the $V_{min\_est}(k)$ with $V_{min\_chip}(k)$, and discard solutions that violate the safe condition in Equation (2.21). Finally, for each $V_{min\_est}(k)$ that satisfies the safe condition, we calculate the average of its resultant $V_{min\_est}(k)$ across k process conditions.

The results in Table 2.13 show that the tunable sensor can achieve a lower supply voltage compared to the normal (non-tunable) ROs in all cases. From the experimental data, we see that the benefits of the tunability vary depending on the difference between V_{min_est} and V_{min_chip}. For example, Figure 2.29 shows that the V_{min_est} values obtained from the non-tunable ROs are very close to the V_{min_chip} values, especially for the *SPRARC_FPU* and *SPARC_MUL* modules.



(a) *SPARC_FPU* (b) *SPARC_TLU* (c) *SPARC_MUL*

Figure 2.29: Distributions of $(V_{min\_est} - V_{min\_chip})$ for different circuit modules. The results show that $(V_{min\_est} - V_{min\_chip})$ is always positive. This implies that the tunable ROs can be used for voltage scaling without causing any timing violations.

Table 2.13: V_{min_est} reduction enabled by the tunability of PVS ROs.

	V_{dd} reduction		Mean
	Average (mV)	Maximum (mV)	$V_{min\_chip}(mV)$
SPARC_FPU	2.7	16.8	851
SPARC_TLU	13.3	31.3	840
SPARC_MUL	2.7	16.8	851

Thus, there is not much room left in which to reduce V_{dd} without causing a timing violation. When V_{min_est} is larger than V_{min_chip}, we can recover the wasted voltage margin by tuning the configurations of PVS ROs. Figure 2.30 shows that by tuning the configuration of the PVS ROs, we can obtain a more aggressive AVS configuration for voltage reduction. For the maximum voltage reduction configuration shown in the figure (green color), we can achieve about $13mV$ voltage reduction compared to the non-tunable ROs, on average (mean of 100 Monte Carlo samples). Note that the voltage reduction varies depending on the process variation. For example, the maximum V_{min} reduction compared to the non-tunable ROs is $31.3mV$ for a specific instance.

In summary, our experimental results confirm that our methodology allows selection of standard cells to build ROs with worst-case voltage scaling characteristics, which can be used as performance monitors for AVS. The overhead $(V_{min\_est} - V_{min\_chip})$ of these ROs varies depending on the circuit. Although our study uses single-V_{th} devices, the methodology can be extended to designs with multi-V_{th} devices by having a set of ROs for each V_{th}. Since the V_{min_est} in our methodology is defined by the maximum V_{min_ro} of all ROs, the V_{min_chip} defined by mixed-VT cells will always be less than V_{min_est}.

2.2.5 Conclusions

We have presented a different approach to enable process-aware voltage scaling. In contrast to the conventional monitoring approaches that attempt to track critical paths, we propose to enable process-aware AVS by synthesizing a set of ROs which achieve a worst-case voltage scaling property across different process conditions. Since the ROs always require a relatively higher voltage to meet their target frequencies than that required by critical paths, a closed-loop AVS guided by these ROs will always scale voltage to a (safe) value that is higher than what is needed by the critical paths. Our experimental results also confirm that through detailed analysis of voltage scaling characteristics, we can design ROs for AVS without any information regarding the critical paths or timing performance of a specific design. At the same time, the proposed method could be too pessimistic, and hence we propose circuit design techniques to tune the

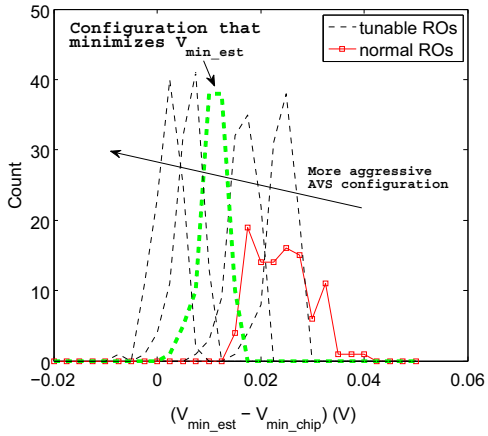


Figure 2.30: Distribution of ($V_{min\_est} - V_{min\_chip}$) for the *SPARC_TLU* testcase with different PVS RO configurations. By tuning the configuration of the ROs, we can change the voltage scaling characteristics (V_{min_est}). An optimized configuration can reduce V_{min_est} by $13mV$ compared to normal ROs.

voltage scaling characteristics of the ROs. We show that the tunability can be used in a scenario where chip frequency is available during ROs characterization. By calibrating the ROs, we can enable up to an additional $30mV$ of supply voltage scaling on a per-instance (per-chip) basis, and up to an average of $13mV$ for a given design. We note that our experiments have been conducted with parameters from a mature ($65nm$) process. The benefit of tunability in the PVS monitors is likely to be larger in less-mature processes which have larger variations around nominal conditions. Intuitively, this is because the voltage scaling characteristics vary more in the presence of process variations. On the other hand, if the cells in a technology have similar voltage scaling characteristics, we would not need the tunability because using the INV-, NAND- and NOR-chained ROs is sufficient to provide the voltage scaling information.

These ROs can also capture circuit delay degradation due to aging mechanisms (e.g., bias temperature instability and hot carrier injection) if the ROs have the same activity as the circuits being monitored. We can capture the aging effect by connecting the ROs and circuits to the same power rails such that the ROs and the circuits are turned on and off together. Alternatively, more sophisticated aging sensors can be used to quantify the additional voltage margin to guardband for circuit aging [112].

2.3 Back-End-Of-Line Layout Optimization for Improved Reliability

Time-dependent dielectric breakdown (TDDB) is becoming a critical reliability issue, since the electric field across the inter-metal dielectric increases as technology scales. Moreover, dielectric reliability is aggravated when interconnect spacings vary due to via and/or wire mask misalignment. Although dielectric reliability can be improved by a larger interconnect pitch, such a guardband leads to significant area overhead.

In this section, we propose to improve dielectric reliability through a post-layout optimization. In the layout optimization, we locally shave and/or shift a fraction of wire width to increase the spacing between wires, and/or between adjacent-layer vias and wires. Separately, we also propose a signal-aware chip-level TDDB reliability estimation method which estimates TDDB stress time of interconnects using net signals obtained from a vectorless analysis.

2.3.1 Introduction

Signal levels on adjacent back-end-of-line (BEOL) interconnects induce an electric field (E) across the insulating dielectric. TDDB occurs when the electrically stressed dielectric forms a conducting path between the interconnects. The *dielectric time-to-failure* (t_F) due to TDDB can be empirically modeled as

$$t_F = Ae^{(-\gamma E^w)} = Ae^{(-\frac{\gamma V^w}{S^w})} \tag{2.26}$$

where A is a fitting parameter, γ is the field enhancement factor, V is the voltage difference across the dielectric, S is the spacing between interconnects, and w is a model-dependent scalar. The common values of w are $\{-1.0, 0.5, 1.0\}$, which correspond to the $\{1/E, \sqrt{E}$ and $E\}$ models [15] [44] [56] [160] [183].

Figure 2.31 shows that the spacing and voltage trends projected by the ITRS [234] [235] lead to an increasing electric field as technology scales. Since t_F reduces with an increasing electric field, it is expected that TDDB will be a major reliability concern for BEOL dielectric. Indeed, at the $20nm$ node (sub-$70nm$ local metal pitch) with litho-etch-litho-etch (LELE) double-patterning, TDDB reliability is a primary limiter to further wiring density improvement [13]. Figure 2.32 shows that a 5% spacing increase can improve interconnect lifetime by 20% (in the year 2011) and that the improvement increases as technology scales.[13]

[13]We calculate interconnect lifetime using Equation (2.26) with w = 0.5 and $\gamma = 15.5(cm/MV)^{0.5}$ [130]. The

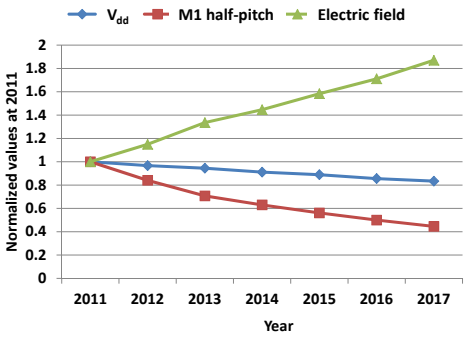


Figure 2.31: Scaling trend of electric field derived from spacing and supply voltage projections [234] [235].

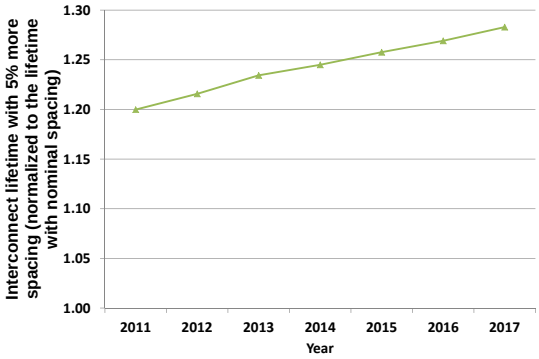


Figure 2.32: Lifetime improvement due to a 5% spacing increase as technology scales.

Recent studies [126] [191] [208] show that mask misalignment between via and wire leads to smaller via-to-wire spacings compared to the wire-to-wire spacings. As a result, the dielectric between a via and wire has a higher electric field and a shorter lifetime. Since the lifetime of a chip is affected by the first dielectric that fails, TDDB reliability improvement should focus on via-to-wire spacings. The study conducted by Xia et al. [208] further clarifies, based on measurement results, that TDDB is dominated by via-to-wire spacing (rather than wire-to-wire spacing). To illustrate the impact of a misaligned via, we simulate the electric field of the dielectric between interconnects using a commercial 3D field solver tool [255]. Figure 2.33 shows that when the via-to-wire spacing is reduced from $70nm$ to $60nm$ due to via misalignment, the electric field around the via is 25% higher than the average electric field between the wires. Moreover, the via-to-wire misalignment is expected to worsen in advanced technology when the vias must land on wires that are misaligned due to LELE double-patterning. Such a worsening TDDB reliability trend will limit the wiring pitch and/or the maximum allowed supply voltage.

values of V and S are obtained from ITRS reports [234] [235].

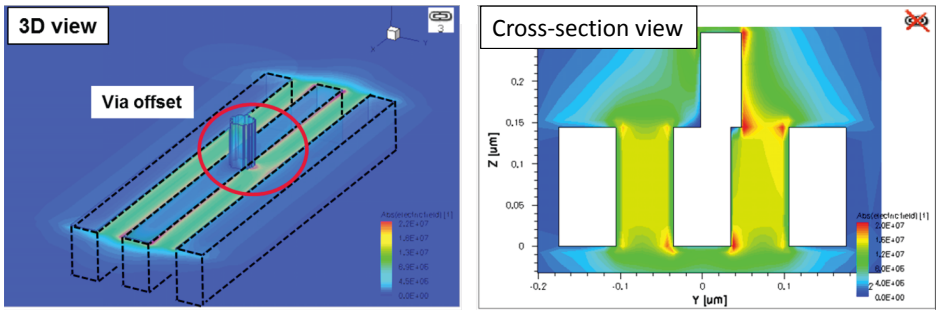


Figure 2.33: Misaligned via reduces the interconnect spacing and enhances the electric field.

To reduce design margin due to BEOL TDDB reliability, processing techniques such as self-aligned via patterning [23] and optimization of etch stop layer [42] have been proposed. We propose alternative approaches to reduce the margin through (1) signal-aware TDDB reliability estimation and (2) post-detailed routing layout optimization. First, conventional TDDB reliability estimation is based on the worst-case assumption in which each interconnect pair is under DC TDDB stress (i.e., each pair of wires always carries opposite logic signals). Such estimation is clearly pessimistic. To reduce the pessimism, we estimate total stress time for interconnects using *state probabilities* (i.e., the probability that an interconnect has a logic state '1') that are available from simulation during the logic design phase of IC implementation. In particular, the state probabilities of all interconnects can be obtained from *electronic design automation* (EDA) tools through vectorless logic simulation [254]. Second, our post-routing optimization improves TDDB reliability by local shifting of the edges of small wire segments to enlarge the particular interconnect spacing (dielectric) that is at risk (see Figure 2.39). Our experimental results in Section 2.3.4 show that this layout optimization has negligible impact on both circuit timing and circuit design, and design-to-manufacturing flows because the layout optimization makes only small changes to segments of wire edges adjacent to vias.

In summary, our contributions are:

- A signal-aware TDDB reliability estimation that reduces pessimism in TDDB reliability analysis.
- A post-routing layout optimization technique to improve TDDB reliability.

2.3.2 TDDB Model

Equation (2.26) is commonly used to describe the relationship between electric field strength and the time-to-failure of a given TDDB test structure. To determine the lifetime of a

chip, we must account for the chip area vulnerable to TDDB as well as the statistics of TDDB. We use the chip-level TDDB model developed by Bashir and Milor [10] and extend it to include the effect of via misalignment as well as different stress time among the interconnects with small via-to-wire spacings.

Chip-Level TDDB Model

Under the same electrical field, identical dielectric may break down at different times. The statistics of dielectric breakdown time can be described by the Weibull or log-normal distributions [25] [47]. Chen et al. show that the Weibull distribution fits (large-sample-size TDDB measurement) data better than the log-normal distribution [47]. Therefore, we use the Weibull distribution to describe the statistics of breakdown time and model the failure rate of a dielectric between interconnects i and j as [10] [47]

$$F_{i,j}(t) = 1 - exp(-(\frac{t}{\eta_{i,j}})^{\beta}) \tag{2.27}$$

where β is the shape factor of the Weibull distribution, t is the total stress time of the dielectric, $F_{i,j}(t)$ is the probability of the dielectric breaking down before time t, and $\eta_{i,j}$ is the *characteristic lifetime* of the dielectric, i.e., the total stress time until 63.2% of the dielectric samples fail. Given a via-to-wire test structure [208], the failure probability of the test structure ($F_{ref}(t)$) can be modeled as

$$\begin{aligned} F_{ref}(t) &= 1 - exp(-(\frac{t}{\eta_{ref}})^{\beta}) \\ \eta_{ref} &= A \cdot exp(\frac{-\gamma V^w}{S_{ref}^w}) \end{aligned} \tag{2.28}$$

where η_{ref} is the characteristic lifetime of the test structure, w is the scalar of a TDDB model and S_{ref} is the via-to-wire spacing. Since the via-to-wire spacings in a chip can be different from that in the test structure, we apply Poisson area-scaling law to model chip-level TDDB reliability [10]:

$$\begin{aligned} F_{i,j}(t) &= 1 - exp[-(\frac{t}{\eta_{i,j}})^{\beta}] \\ &= 1 - exp[-(\frac{t}{A \cdot exp(-\gamma V^w / S_{i,j}^w)(L_{ref}/L_{i,j})^{1/\beta}})^{\beta}] \\ &= 1 - exp[-(\frac{t}{(L_{ref}/L_{i,j})^{1/\beta} \cdot \eta_{ref} \cdot exp(-\gamma V^w (S_{i,j}^{-w} - S_{ref}^{-w}))})^{\beta}] \\ &= 1 - exp[-(\frac{t}{\eta_{ref}\zeta_{i,j}})^{\beta}] \\ \text{where } \zeta_{i,j} &= (L_{ref}/L_{i,j})^{1/\beta} \cdot exp(-\gamma V^w (S_{i,j}^{-w} - S_{ref}^{-w})) \end{aligned} \tag{2.29}$$

Here, we use $S_{i,j}$ and $L_{i,j}$ to define the *critical dielectric area* in between via-wire pairs that is vulnerable to TDDB reliability risk. As shown in Figure 2.34, W_j denotes the width of the j^{th} wire segment, $S_{i,j}$ is the spacing between the i^{th} via and the j^{th} wire, and $L_{i,j}$ is the length of the critical dielectric area in between the via-wire pair . We define L_{via} (resp. W_{via}) as the dimensions of a rectangular via in the preferred (resp. non-preferred) routing direction in the corresponding via layer. We only consider square vias; therefore, L_{via} is always the same as W_{via}. Since the via can be misaligned in the direction parallel to the wire, we extend the length of the critical dielectric area by L_b on each side of the via (in the direction parallel to the wire).

Note that we use several pairs of $S_{i,j}$ and $L_{i,j}$ to represent the critical dielectric area when the area is not rectangular. Similarly, S_{ref} is the via-to-wire spacing in a test structure, and L_{ref} is the total length of the critical dielectric areas in the test structure. We assume the dielectric in test structure is the same as the dielectric in actual chips. Thus, A, β and γ of the chip are the same as those extracted from the test structure.

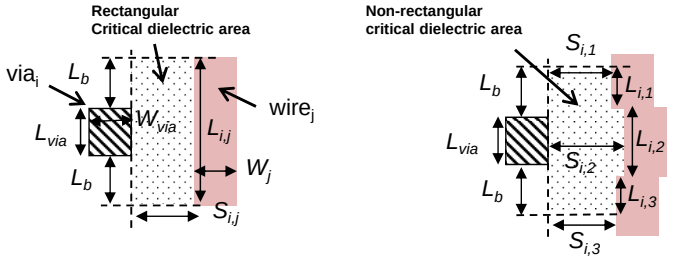


Figure 2.34: Descriptions of geometrical parameters of a via-wire pair.

Equation (2.29) shows that the characteristic lifetime of a dielectric, $\eta_{i,j}$, can be represented in term of test structure characteristic lifetime (η_{ref}) with a scaling factor, $\zeta_{i,j}$. To estimate chip-level failure probability, we apply the weakest link model which defines that a chip malfunctions whenever there is a single failure in any interconnect pair. That is,

$$\begin{aligned} F_{chip}(t) &= 1 - \prod_{i,j} exp(-(\frac{t}{\eta_{i,j}})^{\beta}) \\ &= 1 - exp(-(\frac{t}{\eta_{ref}} \cdot \sum_{i,j} \zeta_{i,j}^{-1})^{\beta}) \end{aligned} \tag{2.30}$$

where F_{chip} denotes the chip-level failure probability.

Signal-Aware TDDB Analysis

Note that the (F_{chip}) Equation (2.26) implicitly assumes that the dielectric is under DC stress, i.e., interconnects around the dielectric always have opposite logic signals. This assumption is clearly pessimistic because interconnect pairs in a chip may not always be stressed. To reduce the pessimism, we model that an interconnect pair is being stressed only when the interconnects have opposite signals. The chip-level failure probability that accounts for actual stress time is given as

$$F_{chip}(t) = 1 - exp(-(\frac{t}{\eta_{ref}} \cdot \sum_{i,j} r_{stress\_i,j}\zeta_{i,j}^{-1})^{\beta}) \tag{2.31}$$

where $r_{stress\_i,j}$ is the ratio of total stressed time between the via i and the wire j to the lifetime of the interconnects.

Although Equation (2.31) is more accurate, extracting the exact stress ratios for all via-wire pairs in a chip is difficult. This is because the logic states of the interconnects (via and wires) are affected by input patterns of the chip, which may be inaccurate or unavailable during chip design time. Even if the input patterns are available, simulating the logic states and extracting the total stress time of all interconnects are time-consuming. To solve the problem of lack of input vectors and slow runtime, we propose to estimate total stress time for interconnects with state probabilities. The state probabilities of all interconnects can be obtained from EDA tools through vectorless logic simulation [254], which is much faster than cycle-by-cycle simulation based on input vectors. Since the state probability only specifies the probability of logic state '1' but not the timing information of the logic state (i.e., when the logic state occurs, and the time duration of the logic state), we assume that the interconnects have the worst-case signal distribution along the time axis, such that the resulting stress time and lifetime estimation is conservative. Given the state probabilities of two interconnects, the worst-case scenario (maximum stress time) is when one interconnect has logic state '1' at the beginning of a period of time and the other interconnect has logic state '0' at the beginning of the same period of time. In this case, the interconnect pair is being stressed at the beginning and at the end of the time period. Based on this observation, we can calculate the worst-case *stress ratio*, $r_{stress\_i,j}$, for each interconnect pair. The *stress ratio* is defined as the fraction of the time when a pair of interconnects have opposite logic signals.

$$r_{stress\_i,j} = \begin{cases} q_i + q_j, & \text{if } = (1 - q_i) > q_j \\ (1 - q_i) + (1 - q_j), & \text{otherwise} \end{cases} \tag{2.32}$$

where q_i is the probability of the i^{th} interconnect to be logic '1'. Estimation of the maximum stress time using Equation (2.32) is illustrated in Figure 2.35. In this example, the logic states of interconnect i (resp. j) over time are "lumped" into a continuous logic "1" signal with a time duration proportional to q_i (resp. q_j). By aligning the signals of interconnects i and j according to the worst case scenario mentioned above, we can estimate the stress ratio using Equation (2.32). We see that the stress ratio obtained by the proposed method is always pessimistic compared to the actual stress ratio.

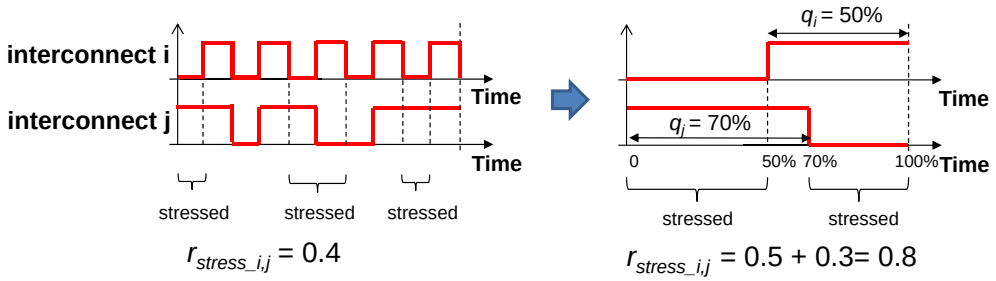


Figure 2.35: Worst-case stress time estimation based on state probabilities.

Modeling Via Misalignment

BEOL via-to-wire spacing can vary due to mask misalignment, lithography-induced spacing variation, etc. To account for the impact of via-to-wire spacing variation on chip-level TDDB reliability, we model the via-to-wire as a normal distribution with zero mean and a standard deviation σ_S. The expectation of $\zeta_{i,j}$ under spacing variation ($\hat{\zeta}_{i,j}$) is given by

$$\hat{\zeta}_{i,j} = \int_{S_{i,j}=S_{i,j}-3\sigma_S}^{S_{i,j}=S_{i,j}+3\sigma_S} P(S_{i,j}) \cdot (L_{ref}/L_{i,j})^{1/\beta} \cdot exp(-\gamma V^w (S_{i,j}^{-w} - S_{ref}^{-w}))dS_{i,j} \qquad (2.33)$$

where $\hat{\zeta}_{i,j}$ denotes the expectation of $\zeta_{i,j}$, and $P(S_{i,j})$ is the probability of the spacing equal to $S_{i,j}$. Since there is no analytical closed-form solution for $\hat{\zeta}_{i,j}$, we approximate it by discretizing the distribution of $S_{i,j}$ into N equal intervals from $S_{i,j} - 3\sigma_S$ to $S_{i,j} + 3\sigma_S$.

$$\hat{\zeta}_{i,j} \approx \sum_{n=1}^{N} \text{cdf}(S_{i,j}(n)) \cdot (L_{ref}/L_{i,j})^{1/\beta} \cdot exp(-\gamma V^w (S_{i,j}(n)^{-w} - S_{ref}^{-w})) \qquad (2.34)$$

Here, $S_{i,j}(n)$ is the n^{th} interval of the discretized $S_{i,j}$, and cdf$(S_{i,j}(n))$ is the corresponding cumulative probability for the n^{th} interval of the discretized $S_{i,j}$.

2.3.3 Post-Route Layout Optimization

Equations (2.29) and (2.31) show that $F_{chip}(t)$ can be reduced by increasing $S_{i,j}$. We therefore propose to improve BEOL TDDB reliability by shifting a small fraction of the wire edges around vias to increase $S_{i,j}$. Note that we want only to make small changes on the wire edges because major layout changes to a routed layout may incur additional design iterations and increase design turnaround time.

Overview

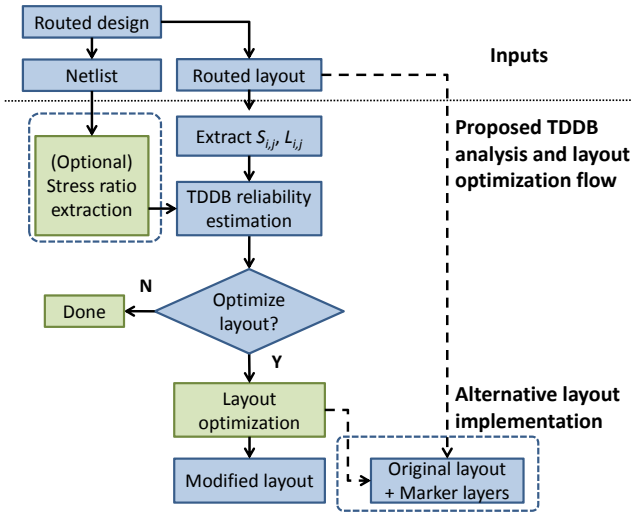


Figure 2.36: Proposed TDDB reliability estimation and layout optimization flow.

Figure 2.36 shows the overview of our layout optimization flow. Given a routed layout, we can extract the via-to-wire spacings $S_{i,j}$ and $L_{i,j}$ to calculate the chip lifetime, t, that corresponds to a failure rate, F_{chip} (e.g., 0.5%). If the design netlist is provided (optimally, with input stimuli), we can also extract the state probabilities to account for the stress ratio between interconnects instead of assuming that the interconnects are always stressed. Based on the results of reliability estimation, a chip designer can decide whether layout optimization is needed. If the designer chooses to apply the layout optimization, the layout optimization will generate an optimized layout in which the via-to-wire spacings are increased. We can also generate marker layers in tapeout GDSII to represent the layout modifications. The marker layers can then be read by an OPC tool flow to shift targeted wire edge locations appropriately during mask data preparation.

Optimizing Layout

Given a routed layout, we collect the via-wire pairs which have via-to-wire spacing smaller than the *safe distance*, S_{safe}. We define S_{safe} as the distance, i.e., spacing, beyond which a dielectric is safe from TDDB (e.g., $S_{safe} \approx 95nm$ in the 32/28nm foundry node with 80nm Mx pitch). We only consider via-wire pairs in which the via is located on the layer above the wire. This is because a via located on the layer below a wire is self-aligned to the wire in a typical dual-damascene process. These self-aligned via-wire pairs have small misalignments and we assume that they are less susceptible to TDDB [99]. For each via-wire pair, we identify *movable* wire edges on each side of the wire segment, such that we can increase the via-to-wire spacing and/or adjust the wire width by shifting the movable edges. As illustrated in Figure 2.37(a), we first define length of the movable wire edges to be the same as the via edge length (L_{via}) and align the movable wire edges to the via edges. Then, we extend each wire edge by L_b at each end point to account for via misalignment in the direction parallel to the wire. Note that the L_b for each end point can be different, to match the magnitude of via misalignment. For example, in Figure 2.37(a), L_b at the top (larger y-coordinate) can be larger than that at the bottom (smaller y-coordinate) if the via misalignment magnitude is larger toward the top compared to the bottom. If movable wire edges are overlapped (see Figure 2.37(b)), we split the movable edges into disjoint, independently movable edges by defining the overlapped region of the edges as new movable edges.

After creating the movable wire edges, we check the vias around the wire segment defined by the movable wire edges. If a via is located in the layer immediately above the wire segment, we do not move the wire edges because moving them may reduce the via landing area, which would lead to lower manufacturing yield. If a via is located in the layer immediately below the wire segment, we can choose to shift the wire edges if the via is self-aligned to the wire in the manufacturing process [23] [99]. With this in mind, we define two layout optimization regimes.

- In Regime 1, we do not shift movable wire edges if a via is located in the layer immediately above or below the layer of the wire segment corresponding to these movable wire edges.
- In Regime 2, we do not shift movable wire edges if a via is located in the layer immediately above the layer of wire segment corresponding to these movable wire edges. We can shift the wire edges if the via is located below the wire segment and there is no via located above the wire segment.

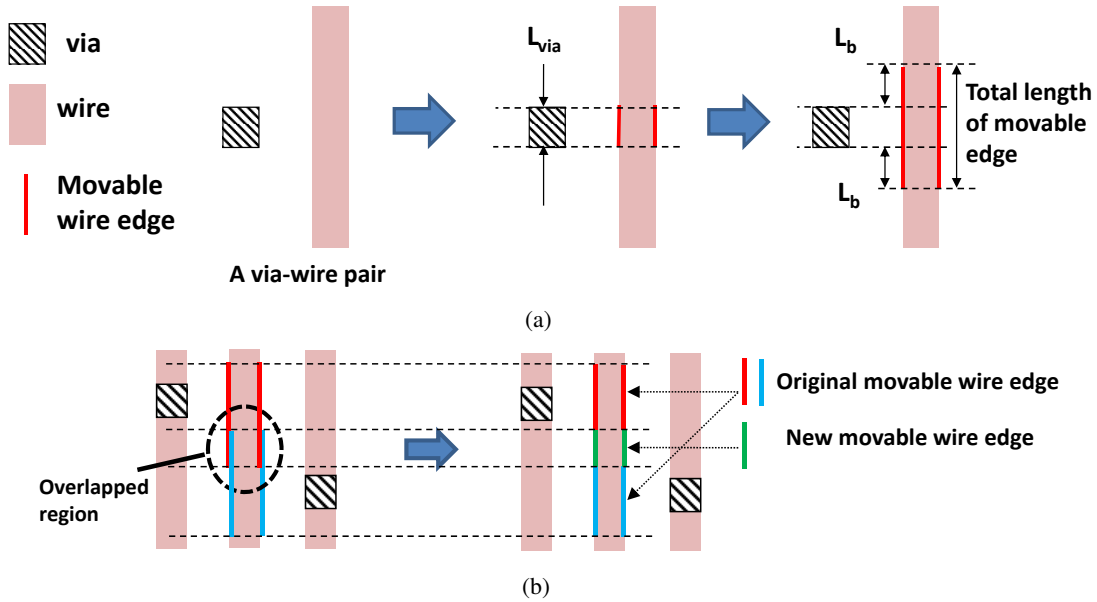


Figure 2.37: Definition of movable edges for cases when (a) there is a via next to a wire (at the layer below the via), and (b) movable wire edges are overlapped (dashed oval on left).

For the remaining movable edges, we apply the following shifting rules. Illustrations of the wire shifting are shown in Figure 2.38.

- If there are vias on both sides of the wire, we shift the movable wire edge inward by ϵ on both sides, to increase via-to-wire spacings.
- If only one side of the wire has vias, we shift the movable wire edge on that side away from the vias by ϵ to increase via-to-wire spacing. We also shift the movable wire edge on the other side by ϵ to preserve wire width.

2.3.4 Experimental Results

Our experiments use four designs {*AES*, *MPEG2*, *JPEG*, *Sparc_EXU*} obtained from the OpenCores [243] and OpenSPARC [242] websites. The designs are implemented using Synopsys 32/28nm NVT, LVT and HVT libraries and BEOL technology files.[14] We synthesize the designs using *Synopsys Design Compiler* [250] and then place and route them using *Cadence SoC Encounter vEDI10.1* [226]. In the experiment setup, we analyze interconnects at layers M2, M3 and M4, which have the same layout parameters. We do not consider interconnects

[14]We have modified the minimum wire width and spacing in the original library exchange format (LEF) file [248] such that minimum width plus minimum spacing is equal to the minimum pitch defined in the LEF.

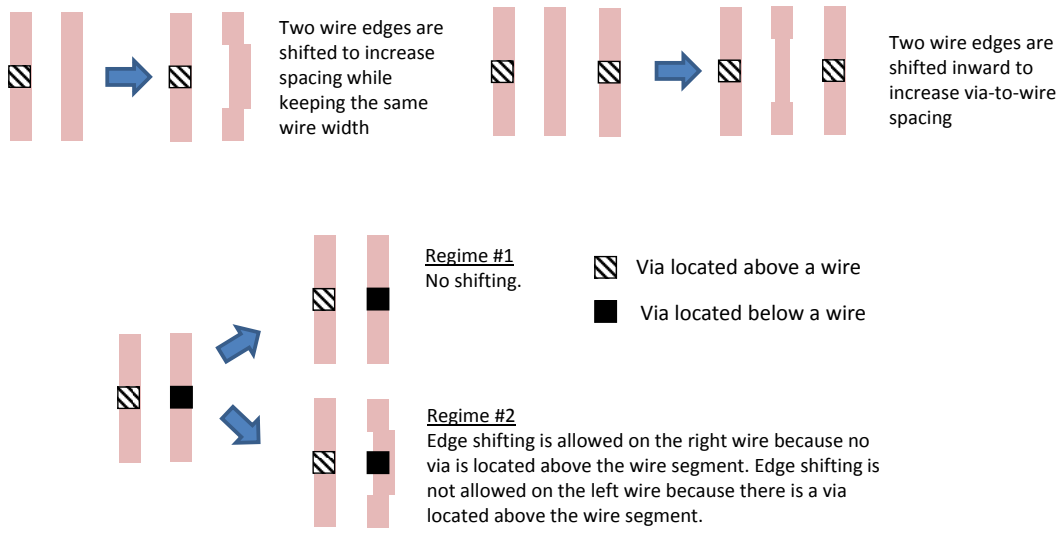


Figure 2.38: Illustrations of wire shifting.

above layer M4 because in this technology they have via-to-wire spacings larger than S_{safe} (i.e., interconnects at layer M4 and above are not vulnerable to TDDB). On the other hand, we do not consider layer M1 because it is used for standard-cell routing, and we assume that the routing in any standard cell is already optimized for TDDB. The parameters of interconnects and related TDDB model parameters are listed in Table 2.14. We assume that σ_S is approximately 3% of the pitch, and define $L_b = 6\sigma_S$. The values β, m, and γ of the TDDB model are obtained from published literature [47] [130]. We fit the values of A, S_{ref} and L_{ref} such that chip lifetime is approximately 10 years. (Although the values of A, S_{ref} and L_{ref} change the TDDB lifetime estimation of a chip, they do not affect the ratio of lifetime estimation of layout optimization compared to the original layout.) We implement the TDDB reliability estimation and layout optimization flow in Figure 2.36 using C++.

Table 2.14: Layout and TDDB model parameters.

Layout parameters	Values	TDDB model parameters	Values
Minimum wire spacing	$80nm$	A	2e17s
Minimum wire width	$80nm$	β	1.0
Minimum via-to-wire spacing	$80nm$	γ	15.5$(cm/MV)^{0.5}$
Via width (L_{via})	$70nm$	w	1.0
σ_S	$5.0nm$	S_{ref}	$80nm$
L_b	$30nm$	L_{ref}	$80nm$
ϵ	$4.0nm$	V	$1.0V$

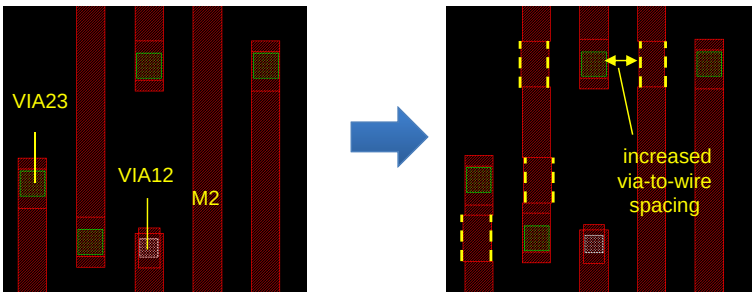


Figure 2.39: Example of BEOL layout modification. The dashed lines indicate the edges of wire segments that are shifted (locally) to increase via-to-wire spacings and improve TDDB reliability.

In our experiment, we apply the layout optimization to each routed layout of the implemented designs. Figure 2.39 shows an example of wires before and after the layout optimization described in Section 2.3.3. In this example, we do not apply edge shifting when there is a via either in the layer immediately above or below the wire segment (defined by the edges). From the figure, we can clearly see that the via-to-wire spacing is increased by shifting the wire edges.

To evaluate the benefits of our proposed methods, we calculate the lifetime, t, of every design by using Equations (2.31), (2.32) and (2.34), with failure rate $F_{chip} = 0.5\%$. For signal-aware TDDB analysis, we extract the state probability of each net obtained from a vectorless analysis [254].[15] Results in Table 2.15 show that by applying our layout optimization method, we can improve chip TDDB lifetime by 9% to 10% (compared to the original layout). The improvement is slightly larger in layout optimization Regime 2, which allows edge shifting whenever there is no via located above the edges. Table 2.15 shows that the lifetime improvements across the two layout optimization regimes differ by only 1%. This means that there are not many movable wire edges that have a via below them.

Table 2.15 also shows that our signal-aware TDDB reliability analysis gives chip lifetime estimates that are 1.7 to 2.8 times the lifetime estimates obtained with a pessimistic DC stress assumption (both estimates obtained without layout optimization). This confirms that TDDB reliability is design-specific, i.e., dependent on the stress ratio of interconnect pairs in the design. In all four designs, we can see a marked reduction of pessimism if we use signal-aware TDDB reliability estimation.

We also study the impact of our layout optimization on BEOL resistance and capaci-

[15] In the vectorless analysis, we assume that all primary inputs have 50% probability to be logic '1'. Based on the extracted state probabilities, we calculate the stress ratios $r_{stress\_i,j}$ for all four designs.

Table 2.15: Chip lifetime (TDDB reliability), normalized to lifetime before layout optimization and with DC stress assumption.

	DC stress			Design-specific stress ratio		
	No opt.	shift edges when there is no via		No opt.	shift edges when there is no via	
		Above or below	Below only		Above or below	Below only
AES	1.000	1.087	1.099	1.696	1.846	1.865
JPEG	1.000	1.085	1.097	2.146	2.333	2.359
MPEG2	1.000	1.087	1.102	2.763	3.017	3.052
SPARC_EXU	1.000	1.089	1.100	1.964	2.138	2.158
Average	1.000	1.087	1.099	2.142	2.334	2.359

tance as well as circuit timing. (1) First, we extract the total changes of resistance (ΔR) and capacitance (ΔC) on each net by extracting the changes in wire width and spacing due to the layout optimization. The third column in Tables 2.16 and 2.17 show that 5.4k (resp. 6.4k) nets are perturbed by the layout optimizations in Regime 1 (resp. Regime 2). This corresponds to approximately 32% (resp. 37%) of the total nets. The results in Tables 2.16 and 2.17 show that the maximum ΔR and ΔC in all the nets in benchmark designs are $< 0.1\Omega$ and $< 0.05fF$, respectively, for both layout optimization regimes. This confirms that our proposed layout optimizations have negligible impact on the wire resistance and capacitance.

(2) Second, we attempt to bound the delay changes due to the layout optimization by analyzing two extreme scenarios. In a *gate-worst* scenario, we add the ΔC of a net to the output pin of the driver cell and do not include any ΔR.[16] Then, we run timing analysis to extract the possible stage delays of the net[17] and calculate the change in delay with respect to each original stage delay without layout optimization. This scenario is designed to estimate the worst-case gate delay impact due to our layout optimization. In a *wire-worst* scenario, we add the ΔC resulting from the layout optimization to the leaf nodes of the net (e.g., input pins of cells driven by the net) and connect the ΔR in series to the output pin of the cell that drives the net. When there is more than one leaf node, we assume that the total ΔC is distributed uniformly among all the leaf nodes. Although this may not be the worst-case setup for wire delay variation, having all ΔR at the output pin and all ΔC at leaf nodes is likely to increase the wire delay variation. By adding up the delay differences of gate-worst and wire-worst scenario, we obtain a pessimistic estimation of delay variation due to the layout optimization. Results in Table 2.16 shows that the maximum Δdelay due to layout optimization in Regime 1 is less than $0.5ps$ for both gate-worst

[16] These changes are made by modifying the original *standard parasitic exchange format* (SPEF) file.

[17] We define the stage delays of a net to be the signal delays of all feasible timing paths from all input pins of the driver cell to all input pins of cells driven by the net.

and wire-worst scenarios. Meanwhile, the average delay variation is less than $0.01ps$ for both scenarios. Similarly, Table 2.17 shows that layout optimizations in Regime 2 also have very small Δdelay for both gate-worst and wire-worst scenarios. Together, in Tables 2.16 and 2.17 show that our layout optimization has negligible timing impact in both Regimes 1 and 2.

Table 2.16: Impact of layout optimization in Regime 1 (no edge shifting when a via is above or below the wire segment).

	#Total Nets	#Opt Nets	Max ΔR (Ω)	Max ΔC (fF)	Worst Δdelay (gate) (ps)		Worst Δdelay (wire) (ps)	
					Max	Average	Max	Average
AES	14k	6.5k	0.037	0.023	0.580	0.010	0.580	0.010
JPEG	29k	7.2k	0.050	0.029	0.263	0.004	0.228	0.004
MPEG2	10k	2.5k	0.054	0.028	0.320	0.005	0.320	0.005
SPARC_EXU	15k	5.5k	0.081	0.041	0.649	0.006	0.850	0.006
Average	17k	5.4k	0.056	0.031	0.453	0.006	0.495	0.006

Table 2.17: Impact of layout optimization in Regime 2 (no edge shifting when a via is above the wire segment).

	#Total Nets	#Opt Nets	Max ΔR (Ω)	Max ΔC (fF)	Worst Δdelay (gate) (ps)		Worst Δdelay (wire) (ps)	
					Max	Average	Max	Average
AES	14k	7.3k	0.037	0.024	0.580	0.010	0.580	0.010
JPEG	29k	8.7k	0.050	0.030	0.263	0.004	0.228	0.004
MPEG2	10k	3.0k	0.070	0.030	0.320	0.005	0.320	0.005
SPARC_EXU	15k	6.4k	0.091	0.041	0.649	0.006	0.850	0.006
Average	17k	6.4k	0.062	0.031	0.453	0.006	0.495	0.006

2.3.5 Conclusions

TDDB is becoming a critical reliability issue for BEOL as technology scales. In the presence of large via-to-wire misalignment, BEOL TDDB limits wire density scaling. To reduce the design margin due to TDDB, we propose a signal-aware chip-level TDDB reliability estimation methodology. Unlike conventional TDDB reliability estimation which assumes that the dielectric is always under DC stress, we estimate the stress ratio based on state probabilities of the routed signal nets in the chip. By using the signal-aware estimation, we show that chip-level TDDB lifetime is approximately twice that obtained from the conventional analysis approach. We also propose a layout optimization method which shifts wire edges to increase via-to-wire spacings to improve BEOL TDDB reliability. Our experimental results using parameters reflec-

tive of the $32nm$ foundry node show that the layout optimization can increase chip-level lifetime by 9% to 10%; impact at $20nm$ and below foundry node is expected to be more substantial. The improvement in chip lifetime also means that the chip can operate at a higher supply voltage with the same lifetime if TDDB is the primary factor that limits the maximum allowed supply voltage.

Our proposed layout optimization method may affect other aspects of the layout such as printability, electromigration, etc. Thus, our ongoing work seeks to include electromigration in the reliability analysis, and to develop a layout optimization method that accounts for both TDDB and EM reliability.

2.4 Acknowledgments

Chapter 2 is in part a reprint of "Synthesis and Analysis of Design-Dependent Ring Oscillator (DDRO) Performance Monitors", *IEEE Transactions on Very Large Scale Integration Systems* (to appear), "DDRO: A Novel Performance Monitoring Methodology Based on Design-Dependent Ring Oscillators", *Proc. International Symposium on Quality Electronic Design*, 2012, "Tunable Sensors for Process-Aware Voltage Scaling", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2012, and "Post-Routing Back-End-of-the-Line Layout Optimization for Improved Time-Dependent Dielectric Breakdown Reliability", *SPIE Advanced Lithography*, 2013.

I would like to thank my co-authors Professor Puneet Gupta, Dr. Kwangok Jeong, Abde Ali Kagalwalla, Andrew B. Kahng and Liangzhen Lai. I also would like to thank Mr. Sorin Dobre for useful conversations.

Chapter 3

Signoff Condition Optimization

This chapter presents various techniques to optimize aspects of signoff, including (i) *operating mode* (i.e., an (operating frequency, voltage) pair), (ii) aging margin, and (iii) back-end-of-line (BEOL) corners. First, we propose a concept of *mode dominance* (see Section 3.1 for the detailed definition) which can be used as a guideline for signoff mode selection. Further, we propose a scalable, model-based adaptive search methodology for signoff mode selection. Second, to optimize the aging margin for a circuit with adaptive voltage scaling (AVS), we study the conditions under which a circuit with AVS requires additional timing margin during signoff. Then, we propose two heuristics for chip designers to characterize an aging-derated standard-cell timing library that accounts for the impact of AVS during signoff. Further, we compare circuits implemented with the aging-aware signoff method based on aging-derated libraries against those based on a *flat timing margin*. Third, to reduce timing margin for BEOL variations, we first analyze the pessimism in the conventional BEOL corner. From observations of the circuit properties of timing-critical paths, we propose a method to identify the paths which can be safely signed off using tightened BEOL corners that embody reduced pessimism.

3.1 Optimization of Overdrive Signoff

In the era of heterogeneous multi-core SoCs, the performance of single-threaded operations limits the overall speedup of applications. Designers use frequency overdrive at elevated voltages to obtain better performance in consumer electronic devices [68]. An operating mode (for simplicity, *mode*) is defined by an (operating frequency, voltage) pair. Devices typically operate at two or three modes, e.g., *supply voltage-scaled* (SVS), nominal and turbo (overdrive).

The nominal and SVS modes correspond to a lower operating voltage and a lower frequency, whereas the overdrive mode corresponds to a higher operating voltage and a higher frequency. Due to limited energy budget, laptops and handheld devices operate at nominal or SVS modes, which we refer to generically as "nominal" in the following, for most of their lifetimes. When high performance is needed to boost CPU-intensive tasks, overdrive mode is turned on for a brief period of operation. The average power consumption (P_{avg}) for a circuit with both nominal and overdrive modes is

$$P_{avg} = r_{OD} \times P_{OD} + (1 - r_{OD}) \times P_{nom} \tag{3.1}$$

where r_{OD} is the *duty cycle* of overdrive mode (i.e., total overdrive time normalized to the total time during which the circuit is turned on) ($0 < r_{OD} < 1$). P_{OD} and P_{nom} are the circuit power in overdrive mode and nominal mode, respectively.

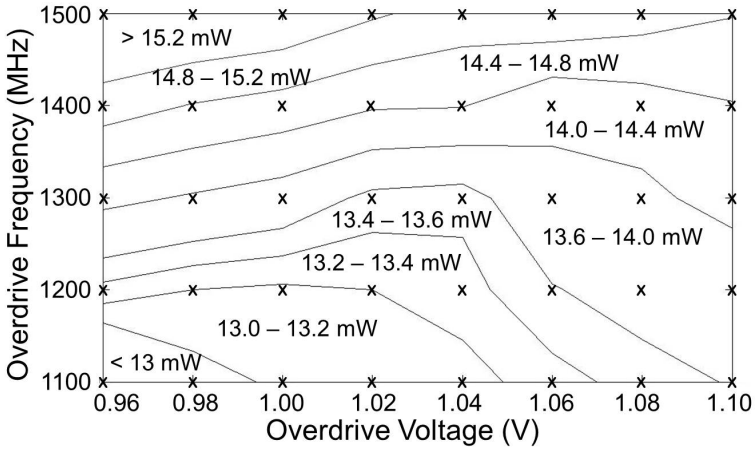


Figure 3.1: Contour plot of P_{avg} versus (frequency, voltage) overdrive signoff corners. Circuit netlist: *AES* [243]. Technology: foundry $28nm$. Nominal mode is (800*MHz*, $0.8V$).

We define the *signoff mode design space* (or *design space*) as the set of combinations of feasible signoff modes. A *point* in this design space specifies m (frequency, voltage) pairs for m-mode signoff, where $m \geq 1$. Signing off at different points in a design space results in circuits with different area, power and performance. For example, Figure 3.1 shows for a testcase implemented in foundry $28nm$ technology that the average power of a circuit can vary by up to 27% across 40 different definitions of the overdrive mode, when the nominal mode is fixed at (800*MHz*, $0.8V$). We assume r_{OD} = 10% in this example, and we note that a different duty cycle r_{OD} would induce a different power contour plot. Even when the overdrive frequency is fixed, Figure 3.2 shows that the average power of a circuit can vary by up to 7% for different overdrive

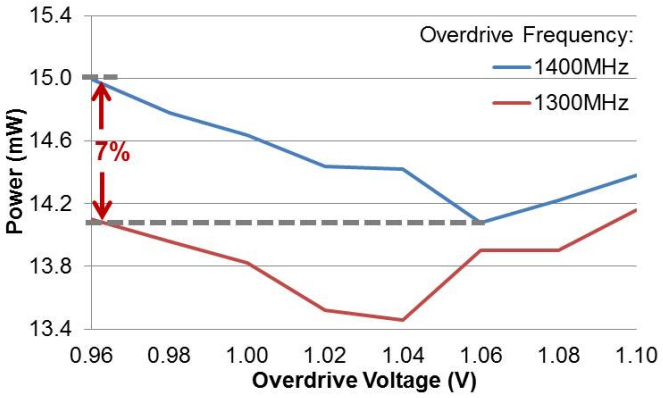


Figure 3.2: P_{avg} versus V_{OD} for fixed f_{OD}.

voltages. Circuit power varies with signoff voltage because when signing off at a lower voltage, buffer insertion to meet timing constraints leads to higher power consumption. On the other hand, although circuit area reduces with higher signoff voltage, power consumption increases with operating voltage. The optimal signoff voltage must comprehend this tension, which is manifested in the unimodal behavior shown in Figure 3.2.

Experimental results in Figure 3.2 suggest that we can reduce design cost by carefully optimizing the signoff modes. Accordingly, we study the *signoff mode optimization problem*, which seeks the optimal nominal frequency (f_{nom}), nominal voltage (V_{nom}), overdrive frequency (f_{OD}) and overdrive voltage (V_{OD}) with respect to optimization objectives and constraints in terms of circuit area, performance and power.

Traditional multi-corner and multi-mode design is conducted by applying a common constraint (e.g., "mission mode") during *synthesis, place and route* (SP&R), and then closing (through netlist and layout optimization steps) and verifying every corner and mode at the signoff stage [177]. Other approaches apply additional margins during physical design or implement incremental optimization for all the corners and modes. However, these approaches can introduce poor timing predictability and be very time consuming. In recent years, EDA tools have offered *Multi-Corner-Multi-Mode* (MCMM) capability [152] [226] [250]. An MCMM methodology simultaneously analyzes and optimizes at all corners and modes of operation throughout the SP&R flow, to obtain improved quality of results (QoR). Applying MCMM throughout the entire SP&R flow can result in better timing convergence at the cost of increased runtime.

The adaptive MCMM flow introduced in [147] identifies and satisfies constraints only at "dominant" modes[18], where a mode is said to be dominant if the circuit implementation is

[18]In [147], dominant modes are defined as the modes that lead to unique or dominant violations.

mainly constrained by the requirements at that mode. In other words, a circuit that satisfies the constraints at dominant modes should also satisfy design constraints at all other modes. By identifying dominant modes, the adaptive MCMM flow reduces runtime and memory usage in IC implementation while retaining similar QoR to optimization at all modes and corners.

A weakness of the adaptive MCMM technology is that it focuses only on the dominant modes *during* implementation. Whenever there is a dominant mode, there can be overdesign at the non-dominant modes. For example, our experimental results in Figure 3.2 show that a circuit implemented to satisfy a dominant mode has up to 7% power consumption overhead for non-dominant modes (i.e., when comparing circuits signed off with overdrive frequency of $1400MHz$, and overdrive voltages of $0.96V$ and $1.06V$). Thus, it is necessary to define the dominant mode *before* implementation. Meanwhile, finding the optimal signoff modes can be very time consuming because the number of (SP&R) iterations using a pure random search grows exponentially with the dimension of design space (i.e., the number of modes) [214].

Another consideration, highlighted in [104], is that lifetime energy consumption can vary widely across different MCMM implementations, depending on the duty cycle of various operating modes. The work of [104] showed that a duty cycle-aware *dynamic voltage and frequency scaling* (DVFS) methodology could save up to 20% lifetime energy over a standard MCMM implementation.

In this section, we propose a method to analyze and identify dominant modes before implementation such that the overdesign resulting from signoff at a dominant mode can be reduced. Moreover, we propose design methodologies to optimize operating mode definitions for multi-mode signoff. A similar multi-mode signoff optimization has been studied by [104]. However, our work achieves greater insight into the basic tradeoff between frequency and voltage at the circuit level; based on this, we propose a more efficient and effective methodology for multi-mode signoff optimization. We furthermore ensure that our approach can comprehend the duty cycle of operating modes, and optimize design signoff accordingly.

Our contributions are summarized as follows.

- We propose a methodology to analyze and identify the dominant modes before circuit implementation.
- We show that for signoff optimization, *equivalent dominance* of all modes should be achieved to avoid overdesign. Based on the property of equivalent dominance, we reduce the runtime of searching for optimal signoff modes by reducing the design space for signoff mode selection.

- We propose a global optimization flow for signoff mode selection which efficiently explores the design space using *model-based adaptive search*. Our proposed methodologies lead to up to 6% performance improvement as compared to the traditional "signoff and scale" method. The signoff modes identified by our proposed flow lead to less than 3% power overheads compared to the optimal result obtained by exhaustive search over all possible combinations of signoff modes.

The following notations are used in this section.

- Signoff frequencies: f_{nom} and f_{OD}
- Signoff voltages: V_{nom} and V_{OD}
- Duty cycle in overdrive mode: r_{OD} $(0 < r_{OD} < 1)$
- Power consumption at two modes: P_{nom} and P_{OD}
- Average power: P_{avg} $(= (1 - r_{OD}) \times P_{nom} + r_{OD} \times P_{OD})$
- Peak power: P_{peak} $(= P_{OD})$

3.1.1 Dominance of Modes

Design Cone

To analyze the dominance of modes, we define the concepts of *mode* and *design cone* as follows.

Definition: The *design cone* of a given mode ***M*** is the union of (maximum frequency, voltage) operating modes for all feasible circuit implementations that are signed off at mode ***M***.

Figure 3.3 illustrates the design cone ***R*** (shaded region) of a nominal mode ***A***. Different circuits can be signed off at mode ***A***, and each of these circuits will have its own frequency versus voltage tradeoff curve. The boundary of the design cone is determined by the upper and lower bounds of the maximum frequencies of circuits that can be achieved at different voltages.

To study the minimum and maximum feasible frequencies at different voltages, we model the corner cases of timing-critical paths in a digital circuit by simulating chained standard cells with different gate types, threshold voltages (V_{th}) and fanouts. We also consider the impact of wire resistance. We use standard cells from a dual-VT $28nm$ commercial foundry

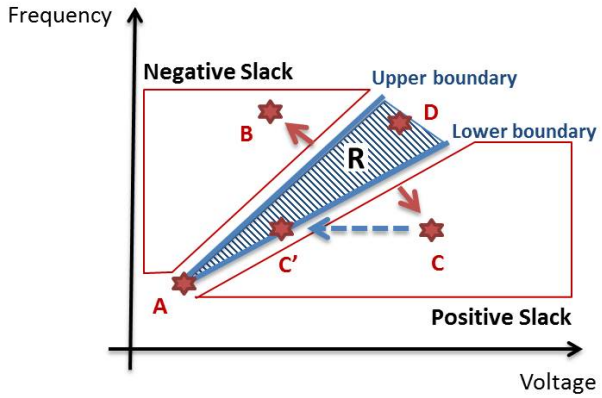


Figure 3.3: Design cone of mode ***A*** (the shaded region). A circuit signed off with mode ***A*** will have negative (respectively, positive) timing slacks when operated at mode ***B*** (respectively, ***C***).

library. The simulation results in Figure 3.4 show that the frequencies[19] of inverter and NOR chains increase essentially linearly as supply voltage increases.[20] Therefore, by approximating the frequency versus voltage tradeoff curves as straight lines [69], we determine the upper and lower boundaries of a design cone by the curves with maximum and minimum slopes.

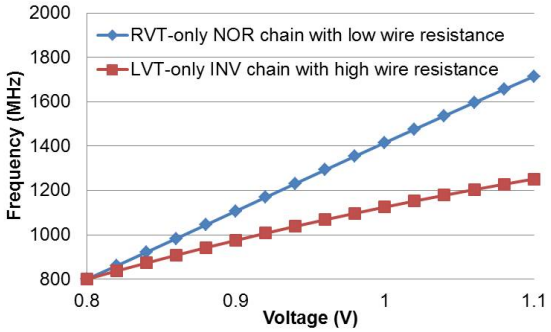


Figure 3.4: Frequency versus voltage tradeoffs for LVT-only inverter chain and RVT-only NOR chain which satisfy the timing constraint ($800MHz$ at $0.8V$).

Data in Table 3.1 shows that the slope of the frequency versus voltage tradeoff is mainly determined by the threshold voltages, gate types of standard cells and wire resistance. Meanwhile, fanout has little influence on the slope of frequency versus voltage tradeoffs. The rightmost column in Table 3.1 shows the cases where per-stage wirelength is maximized with respect to transition constraints from Liberty files. We also observe that circuits with regular threshold

[19] We use the term "frequency" here to indicate the reciprocal of the path delay. This matches our usage in the discussion below of the circuit frequency versus voltage tradeoff.

[20] The simulation results in Figure 3.4 are for X25 size inverters and NOR gates. Our studies indicate that the frequency versus voltage tradeoff trend is not affected by the size of inverters.

voltage (RVT) cells have steeper tradeoff slopes compared to circuits with low threshold voltage (LVT) cells; this is also observed in [149], where in $45nm$ CMOS, the slope of the frequency versus voltage tradeoff is $3\times$ larger for HVT than for LVT cells.

Table 3.1: Slopes of frequency versus voltage tradeoffs for different chained standard cells. Delay = $1.25ns$ (corresponding to frequency = 1/delay = 800*MHz*) at $V = 0.8V$.

V_{th}	Fanout	Slopes (*MHz*/V)			
		Per-stage wirelength = 1 μm			Maximum per-stage wirelength
		INV	NAND	NOR	INV
LVT	4	2115	2312	2331	1498
LVT	16	2047	2259	2299	1958
RVT	4	2766	2917	3066	2077
RVT	16	2685	2835	3016	2619

Note that the delay and supply voltage of a circuit also affect the frequency versus voltage slope. However, a design cone is defined at a mode where the delay (reciprocal of frequency) is fixed. Thus, we estimate the upper and lower boundaries of the design cone using RVT-only NOR chain and LVT-only inverter chain and high wire resistance, respectively. Figure 3.4 shows the estimated design cone for the mode (800*MHz*, $0.8V$).

Dominance

Definition: Given the design cone of mode M_1, a mode M_2 (f_{M_1}, V_{M_2}) has a *positive slack* (respectively, a *negative slack*) with respect to mode M_1 if f_{M_2} is below (respectively, above) the lower (respectively, upper) boundary of design cone at V_{M_2}.

In Figure 3.3, point ***A*** indicates the nominal signoff mode. When another mode (e.g., mode ***C***) is located below the design cone of the signoff mode (e.g., mode ***A***), this is a positive slack. The positive slack can be exploited to either increase the frequency (performance) of mode ***C***, or reduce the operating voltage (power). We say that the existence of positive timing slack indicates *overdesign*.

We illustrate the use of positive slack to reduce power without incurring any penalty in either performance or circuit area, using mode ***A*** and mode ***C*** in Figure 3.3. We select a mode ***C'*** that is located on the lower boundary of the design cone corresponding to mode ***A***. The mode ***C'*** has the same frequency as the mode ***C***. By our definition, a design cone represents all circuits that can be signed off at the corresponding mode. Further, the lower boundary of a design cone

indicates the circuit with the loosest timing constraint. Thus, any circuit signed off at mode ***A*** satisfies timing constraints at mode ***C'***, where circuits signed off with mode ***A*** and mode ***C*** can operate at mode ***C'*** without timing violation.[21] Moreover, mode ***C'*** has lower operating voltage than mode ***C***, which leads to less power consumption, while both have the same performance. Changing the signoff mode from mode ***C*** to mode ***C'*** always leads to power reduction regardless of the duty cycle. Hence, the positive slack can be exploited to reduce power without incurring any penalty in either performance or area.

On the other hand, when a mode (e.g., mode ***B***) is above the design cone of the signoff mode (e.g., mode ***A***), negative timing slack occurs. This is because mode ***B*** has tighter timing constraints than the upper boundary of the design cone. Signing off at mode ***A*** cannot satisfy the timing requirement at mode ***B***. Such negative slack can be eliminated by increasing the operating voltage at mode ***B***.

Definition: Given two modes M_1 and M_2, if mode M_2 shows positive slacks with respect to mode M_1, we define mode M_1 as the *dominant mode*, and mode M_2 as the *dominated mode*.

For example, in Figure 3.3, mode ***A*** is dominant and mode ***C*** is dominated. The dominant mode has tighter constraints, so when constraints of both modes need to be satisfied, the dominant mode determines the properties of a design. Such properties can encompass area, number of instances, total capacitance, slope of the frequency versus voltage tradeoff curve, etc. When neither of two modes is dominant with respect to the other, we say that they demonstrate *equivalent dominance*: their constraints are equivalently strict and the properties of a design are determined by both modes. Moreover, we expect that such properties are similar to those of the design that is signed off at either of the two modes individually. In Figure 3.5, modes ***A*** and ***B*** exhibit equivalent dominance.

Definition: Given two modes M_1 and M_2, when mode M_1 is in the design cone of mode M_2 and mode M_2 is in the design cone of mode M_1, we say that mode M_1 and mode M_2 exhibit equivalent dominance.

Based on the equivalent dominance concept, we state the following.

Lemma 1: If two modes do not exhibit equivalent dominance, then each mode is outside of the design cone of the other mode.

Proof (by contradiction): Assume toward a contradiction that the claim is false, i.e., modes M_1 and M_2 do not exhibit equivalent dominance, but one mode (M_1) is located in the design cone of the other (M_2). According to the definition of design cone, any point in the design cone of

[21] We only consider setup timing constraints in this study.

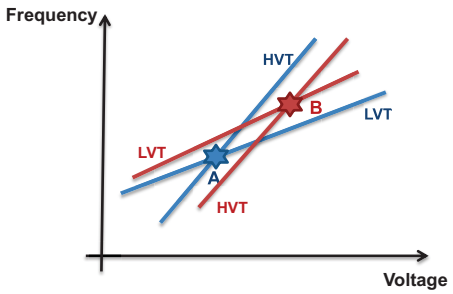


Figure 3.5: Modes ***A*** and ***B*** exhibit equivalent dominance, where each is in the other's design cone.

M_2 lies on a frequency versus voltage tradeoff curve corresponding to a circuit signed off at M_2. Therefore, there is at least one circuit with a frequency versus voltage tradeoff curve that passes through both M_1 and M_2. This means that M_2 is also in the design cone of M_1. Hence, modes M_1 and M_2 exhibit equivalent dominance, contradicting our initial assumption. □

Lemma 2: Multi-mode signoff at modes which do not exhibit pairwise equivalent dominance leads to overdesign.

Proof: If a set of modes does not exhibit pairwise equivalent dominance, then there exist two modes for which equivalent dominance does not hold. According to *Lemma 1*, neither mode is located in the design cone of the other. Then, one of the modes must be dominant, and the other dominated. By definition of a dominated mode, the circuit being implemented at the dominated mode will have positive timing slack. Regardless of the duty cycle, positive timing slack indicates overdesign (cf. Figure 3.3). Therefore, at least one mode will be overdesigned if a set of modes does not exhibit pairwise equivalent dominance. □

Lemma 3: Mutual pairwise equivalent dominance among $m \geq 3$ modes requires that the modes are collinear in the (V, f) space for signoff.

Proof (by induction on m)**:**

Base Case (m = 3). Per the discussion in Section 3.1.1, the frequency versus voltage tradeoff curve for a given circuit is taken to be a straight line. Further, any one circuit implementation corresponds to only one frequency versus voltage tradeoff curve. Thus, signoff with any two out of the three modes will determine a frequency versus voltage tradeoff curve (corresponding to the resultant circuit). Whenever the third signoff mode is below the frequency versus voltage tradeoff curve of the other two modes, the supply voltage can be reduced to achieve lower power and still meet timing constraints; this corresponds to overdesign. And, whenever the third mode is above the frequency versus voltage tradeoff curve of the other two modes, there must be a

timing violation at the third mode; this corresponds to a failed design. Therefore, the third signoff mode must be on the frequency versus voltage tradeoff curve of the other two modes (i.e., the three modes are collinear) for equivalent dominance to hold.

Inductive Step. As the induction hypothesis, assume that when any k modes ($k \geq 3$) exhibit mutual pairwise equivalent dominance, they are collinear in the design space. We wish to prove that any (k+1) modes with mutual pairwise equivalent dominance must be collinear. Pick any subset G_1 of k modes and let A be the remaining $(k+1)^{st}$ mode. The modes in G_1 are collinear. Pick any subset G_2 of k modes that includes A. The modes in G_2 are collinear. Since $|G_1 \cap G_2| \geq 2$ all $(k+1)$ modes are collinear. □

Figure 3.6 illustrates an example where four modes exhibit equivalent dominance. Line ***D-A-B-C*** is the desired design space for signoff (i.e., without incurring overdesign). We note that marketing or other product requirements may well lead to multiple modes that are not collinear in the design space. In such a situation, there must be overdesign with respect to at least one of the modes. A methodology to define signoff modes to minimize some global measure of overdesign is beyond our present scope, and we focus on scenarios involving just two modes in our work.

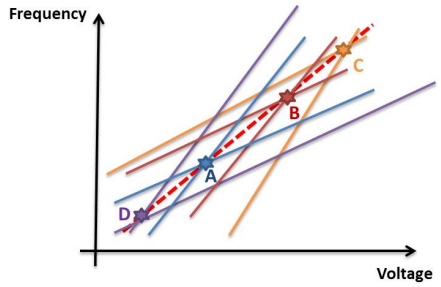


Figure 3.6: Four modes exhibit equivalent dominance. The desired design space is the line ***D-A-B-C***.

3.1.2 Problem Formulations

To sign off a circuit that operates at both nominal and overdrive modes (with a given duty cycle r_{OD}, and constraints on power and supply voltages), we need to select four parameters: f_{nom}, V_{nom}, f_{OD} and V_{OD}.

Definition: We define the problem where m parameters are given, and n parameters must be determined, as the m+n *problem*. In particular, we are interested in cases where $m + n = 4$, and $m = 0, 1, 2, 3$.

The $3+1$ Problem

We classify the $3+1$ problem into two types. The first type, where two frequencies and one voltage are given, is a common scenario in typical IC design flows, e.g., for mobile application processors. This is because f_{nom} and V_{nom} are usually defined by the technology node, and f_{OD} is usually determined by the (market-driven) product specification. Since the performance at both modes is predefined, the objective in this kind of problem can be minimization of power consumption or area. In light of package and reliability requirements, the maximum operating voltage and the peak power consumption are usually set as constraints. In the second type, two voltages and one frequency are given, and we search for the unknown frequency in the signoff optimization.

The $2+2$ Problem

There are four types of $2+2$ problems: (I) given one mode, search for the other mode; (II) given two frequencies, search for signoff voltages; (III) given two voltages, search for signoff frequencies; and (IV) given a voltage at one mode and a frequency at the other mode, search for the other two parameters. Type IV is not a practical use model because the operating voltage at one mode is unrelated to the frequency at the other mode.

In our work, we study the following $2+2$ problems.[22]

The FIND_OD Problem (Type I):

Inputs: f_{nom}, V_{nom} and duty cycle (r_{OD})
Objective: maximize f_{OD}
Constraints: $P_{peak} \leq C_1$; $P_{avg} \leq C_2$; $V_{OD} \leq C_3$
Outputs: f_{OD} and V_{OD}

The FIND_NOM Problem (Type I):

Inputs: f_{OD}, V_{OD} and r_{OD}
Objective: maximize f_{nom}
Constraints: $P_{peak} \leq C_1$; $P_{avg} \leq C_2$
Outputs: f_{nom} and V_{nom}

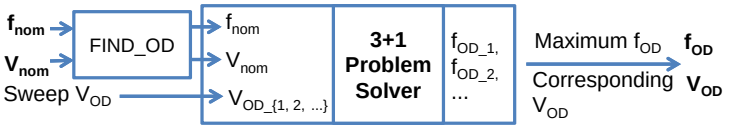
[22]To our knowledge, the $1+3$ problem would not occur in a real product design context. Moreover, it could be solved by sweeping one parameter at a time and optimally selecting the other two parameters (i.e., reducing to the $2+2$ problem). The $0+4$ problem is also not a practically relevant formulation. Therefore, we do not study these problems.

The FIND_VOLT Problem (Type II):

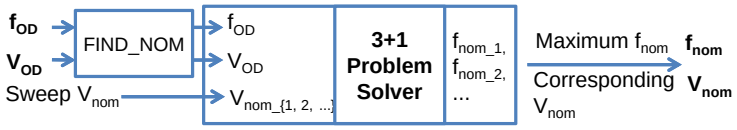
Inputs: f_{nom}, f_{OD} and r_{OD}
Objective: minimize P_{avg}
Constraints: $P_{peak} \leq C_1$; $V_{OD} \leq C_2$
Outputs: V_{nom} and V_{OD}

The FIND_FREQ Problem (Type III):

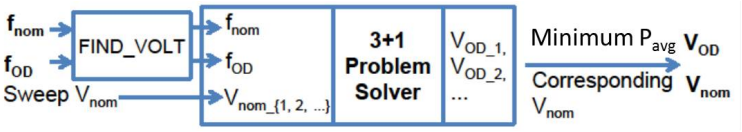
Inputs: V_{nom}, V_{OD} and r_{OD}
Objective: maximize $(1 - r) \times f_{nom} + r \times f_{OD}$
Constraints: $P_{peak} \leq C_1$; $P_{avg} \leq C_2$; $V_{OD} \leq C_3$
Outputs: f_{nom} and f_{OD}



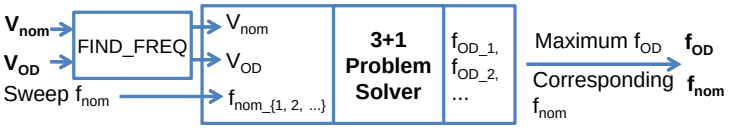
(a) Reduction from FIND_OD to a $3 + 1$ problem.



(b) Reduction from FIND_NOM to a $3 + 1$ problem.



(c) Reduction from FIND_VOLT to a $3 + 1$ problem.



(d) Reduction from FIND_FREQ to a $3 + 1$ problem.

Figure 3.7: Reductions from $2 + 2$ problems to $3 + 1$ problems.

The $2+2$ problems can always be reduced to $3+1$ problems by sweeping one unknown parameter. Figure 3.7 illustrates the reduction relationships. The FIND_OD problem is reduced to the $3 + 1$ problem by sweeping V_{OD}. A range of V_{OD} values, together with given f_{nom} and V_{nom}, are fed into the $3 + 1$ problem solver. Among the corresponding output f_{OD} values,

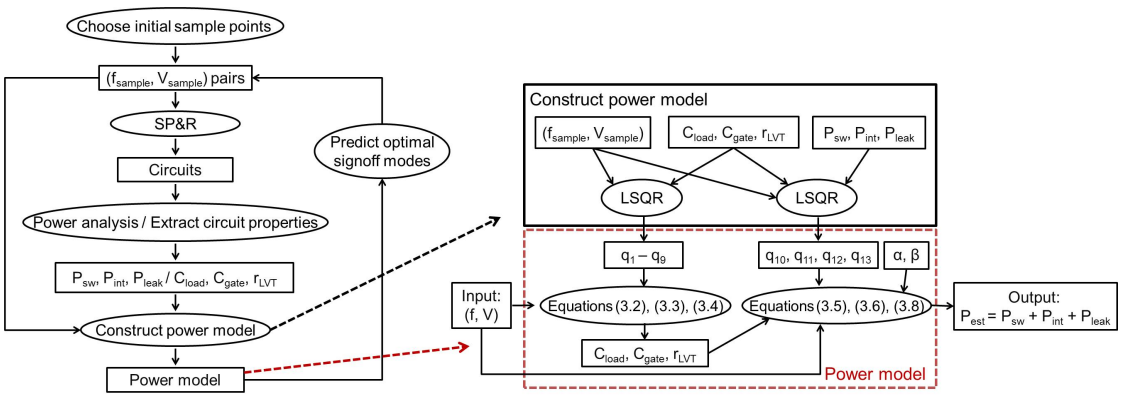


Figure 3.8: Our power model is constructed based on initial samples (obtained by executing SP&R). The left flow chart shows the proposed adaptive search, where we iteratively sample (run SP&R) and update the power model. The dotted box shows our power model.

the one that offers the highest performance is selected as the solution of the FIND_OD problem. Similarly, the FIND_NOM problem can be reduced to a $3{+}1$ problem by sweeping V_{nom}. For the FIND_VOLT problem, where two frequencies are given, one can sweep either V_{nom} or V_{OD}. If we sweep V_{nom}, then from the outputs of the $3{+}1$ problem, we select the V_{OD} and corresponding V_{nom} that offer minimum average power consumption (taking the given duty cycle r_{OD} into account) as the output of the FIND_VOLT problem. For the FIND_FREQ problem, where two signoff voltages are given, one can sweep either f_{nom} or f_{OD}. If we sweep f_{nom}, then from the outputs of the $3 + 1$ problem, we select the f_{OD} and corresponding f_{nom} that offer the highest performance as the output of the FIND_FREQ problem.

3.1.3 Efficient Exploration of the Signoff Mode Design Space

The key challenge in signoff mode optimization is to efficiently search for the desired modes using a small number of implementation trials. To this end, we propose a *model-based adaptive search* to explore the design space for signoff mode selection. In the model-based adaptive search, new solutions are determined using models which are updated or derived from implementations with previous solutions [89]. Figure 3.8 shows our adaptive search flow. We construct our power model based on initial samples. Using the power model, we *predict* the optimal signoff mode and sample (i.e., run SP&R) at the predicted mode. We iteratively sample and update the power model until the flow converges.

Power Model

Based on power at the nominal mode (P_{nom}) and at the overdrive mode (P_{OD}), we use Equation (3.1) to calculate the average power (P_{avg}). In the following discussion, we focus on the construction of our power model at a single mode. The dotted box in Figure 3.8 shows our power model. Following industry-standard models (Liberty [237]) and tools (e.g., *PrimeTime* [254]), we model circuit power as being comprised of three components – switching (P_{sw}), internal (P_{int}) and leakage (P_{leak}). Our power model uses the following circuit properties: load capacitance (C_{load}), which includes wire capacitance and the capacitance of input pins driven by nets [105]; total gate capacitance (C_{gate}); and percentage of cell instances with different V_{th} flavors ($r_{\{LVT,RVT\}}$). As we previously observed in Figure 3.2, circuit power exhibits unimodal behavior with varying signoff voltage. This suggests that we may model power as a second-order polynomial of the signoff voltage. We also observe below that power linearly depends on circuit properties. Therefore, we also model the circuit properties as second-order polynomials of the signoff voltage or frequency, as

$$c_{load} = q_1 \times V^2 + q_2 \times V + q_3 \tag{3.2}$$

$$c_{gate} = q_4 \times V^2 + q_5 \times V + q_6 \tag{3.3}$$

$$r_{LVT} = q_7 \times V^2 + q_8 \times V + q_9 \tag{3.4}$$

where $q_1, ..., q_9$ are fitting parameters. Equations (3.2), (3.3) and (3.4) are used when V is the variable in adaptive search; when f is the variable in adaptive search, we use f in place of V in Equations (3.2), (3.3) and (3.4).

Note that circuit properties may not always behave according to second-order relationships with the signoff voltage or frequency, and that this can lead to errors in power estimation. However, our experimental results (see Section 3.1.5) show that the estimation error is less than 10%. We use the estimated circuit properties from Equations (3.2), (3.3) and (3.4) to model power components. Details are given as follows.

Net switching power is the power dissipated by charging and discharging the load capacitance during operation. We model net switching power as

$$P_{sw} = q_{10} \times \alpha \times C_{load} \times f \times V^2 \tag{3.5}$$

where α is the switching activity factor; f and V are operating frequency and supply voltage, respectively; and q_{10} is a fitting parameter used during adaptive search.

Internal power includes the short circuit power and the power dissipated by charging and discharging the gate internal capacitance. A basic model due to Veendrick [198] indicates that the short circuit power is proportional to the switching activity factor and frequency. Nose and Sakurai [161] improve the model for advanced CMOS technology and show that short circuit power of a cell is proportional to supply voltage and input node capacitance. Since the internal power mainly consists of the short circuit power (at least, as characterized in the production library models that we work with), we model the internal power as

$$P_{int} = q_{11} \times \alpha \times C_{gate} \times f \times V^2 \tag{3.6}$$

where q_{11} is a fitting parameter used during adaptive search.

Leakage power is mainly composed of subthreshold leakage and gate-tunneling leakage. Previous work shows that the leakage power exhibits an approximately linear relation with total transistor width [46]. But, [46] considers only subthreshold leakage (and presumably does not consider the use of multi-channel length libraries). The work of [121] uses the number of cell instances to model leakage power, and reports approximately 98% accuracy with respect to the leakage power reported by a commercial tool. Our experiments, however, show that estimation with gate capacitance is more accurate than when only using the number of cell instances. Further, extracting gate capacitance rather than transistor width is more practical during circuit implementation. Therefore, we use gate capacitance as a parameter to fit leakage power. Since subthreshold leakage current depends exponentially on supply as well as threshold voltages, we use the functional form $e^{\theta \times V}$ (θ is a parameter depending on technology and threshold voltages of transistors) to model the leakage current. To model the impact of cell instances differing in threshold voltage flavors on leakage power, we use percentages of LVT and RVT cells in the model. Note that we do not consider channel length biasing in our present work, but it can be taken into account in the same way that we handle multiple V_{th} flavors. Our model for leakage power is given as

$$P_{leak} = V \times C_{gate} \times (r_{LVT} \times e^{\beta \times V} + r_{RVT} \times e^{\gamma \times V}) \tag{3.7}$$

where β and γ are coefficients used to fit the relationship between supply voltage and leakage current for different V_{th} flavors; and $r_{\{LVT,\ RVT\}}$ are percentages of LVT and RVT cell instances, respectively. We observe that impacts of RVT cells on P_{leak} are quite small. Therefore,

we simplify the equation for leakage power estimation as

$$P_{leak} = V \times C_{gate} \times (q_{12} \times r_{LVT} + q_{13}) \times e^{\beta \times V} \tag{3.8}$$

where q_{12} and q_{13} are fitting parameters used during adaptive search.

We emphasize that Equations (3.5), (3.6) and (3.8) are not for accurate power calculations. Rather, these equations are based on chosen parameters for power estimation within our adaptive search methodology.

Given the actual power value P_{act} of an implemented design, we define the *accuracy* of the corresponding estimated power value P_{est} (from our power model) as $(1 - |P_{act} - P_{est}|/P_{act})$. By using our model, we achieve approximately 97% accuracy with our implementations. In a multi-mode signoff, since the circuit is mainly determined by the dominant mode, which has the tightest timing constraints, we extract the properties of the circuit implemented at the dominant mode to model C_{load}, C_{gate} and r_{LVT}. However, when two or more modes exhibit equivalent dominance, we choose the modes that are not yet fixed and among these modes we choose the mode with the largest duty cycle for power modeling since it has the greater impact on P_{avg}.

Adaptive Search

We now propose two generic adaptive search flows for signoff mode selection. We then extend them to solve the $3 + 1$ and $2 + 2$ problems described in Section 3.1.2.

Given a signoff frequency (f), we use the MIN_POWER flow in Algorithm 1 to search for the signoff voltage (V) that minimizes circuit power (P). In Algorithm 1 (MIN_POWER), inputs V_{min} and V_{max} are user-specified minimum and maximum signoff voltages, respectively. V_{stop} is a stopping criterion for adaptive search. In Line 1, we run SP&R at modes (f, V_{min}), (f, V_{max}) and (f, $(V_{min} + V_{max})/2$). Then, in Lines 2–3, we extract the circuit power and circuit properties from the implemented circuits, and fit the coefficients q_{10}, q_{11}, q_{12} and q_{13} in Equations (3.5), (3.6) and (3.8). In Line 6, based on the power model obtained in Line 3, we predict the optimal signoff voltage to minimize power. We then run SP&R with the predicted signoff voltage and extract circuit information to update the power model in Lines 7–9. If the change in the value of the estimated optimal signoff voltage is less than V_{stop}, the adaptive search terminates. Otherwise, more accurate estimation of the optimal signoff voltage is predicted from the improved power model.

Given a signoff voltage (V), we use the MAX_FREQ flow in Algorithm 2 to search for the maximum signoff frequency (f) under particular power constraints.

Algorithm 1 Adaptive search for the optimal V to minimize P.

Procedure MIN_POWER
Inputs : f, V_{min}, V_{max} and V_{stop}
Output : V
1: run SP&R with (f, V_{min}), (f, V_{max}), (f, $\frac{V_{min}+V_{max}}{2}$)
2: extract circuit information
// circuit information includes C_{load}, C_{gate}, r_{LVT}, P_{sw}, P_{int} and P_{leak}
3: build the power model based on extracted information
4: $i \leftarrow 1$; $V_0 \leftarrow -\infty$
5: **while** $\Delta V \geq V_{stop}$ **do**
6: $V_i \leftarrow$ select the optimal V based on the power model
7: run SP&R with (f, V_i)
8: extract circuit information
9: update the power model using least squares regression based on extracted information
10: $\Delta V \leftarrow V_i - V_{i-1}$
11: $i \leftarrow i + 1$
12: **end while**
13: **return** V_{i-1}

In Algorithm 2, f_{min} and f_{max} define the range of signoff frequency selection, where f_{min} is the predefined lower bound on performance, and f_{max} is the maximum achievable frequency with voltage V. Algorithm 2 builds and updates the power model similarly to Algorithm 1, but seeks a maximum achievable frequency under the given power constraints.

3.1.4 Methodology

In MCMM methodology, all mode-corner combinations must be analyzed during implementation. Thus, execution time of MCMM SP&R is significantly slower than with a single signoff mode [177]. The design space for signoff increases exponentially with the number of signoff modes. Thus, exhaustive search for the optimal signoff modes (e.g., by implementing circuits with MCMM methodology at many combinations of modes in a design space) is infeasible. We reduce the design space for signoff mode selection based on the concept of equivalent dominance described in Section 3.1.1. According to *Lemma 2*, signing off circuits at modes that are not equivalently dominant will lead to overdesigned circuits. Thus, we search only the signoff mode design space in which the equivalent dominance property holds; this is much smaller than the entire design space. Note that for variant duty cycles, the design cone remains the same and *Lemma 2* still holds. Therefore, with any duty cycle, we must still select signoff modes that exhibit equivalent dominance to avoid overdesign. However, within the design cone, for fixed nominal and overdrive modes, different duty cycles lead to different average power, and optimal solutions for signoff mode selection can be different. Our power model estimates P_{avg} based

Algorithm 2 Adaptive search for the maximum f under power constraint P_{max}.

Procedure MAX_FREQ
Inputs : V, P_{max}, f_{min}, f_{max} and f_{stop}
Output : f
1: run SP&R with (f_{min}, V), (f_{max}, V), ($\frac{f_{min}+f_{max}}{2}$, V)
2: extract circuit information
// circuit information includes C_{load}, C_{gate}, r_{LVT}, P_{sw}, P_{int} and P_{leak}
3: build the power model based on extracted information
4: $i \leftarrow 1$; $f_0 \leftarrow -\infty$
5: **while** $\Delta f \geq f_{stop}$ **do**
6: $f_i \leftarrow$ select f based on the power model such that $P = P_{max}$
7: run SP&R with (f_i, V)
8: extract circuit information
9: update the power model using least squares regression based on extracted information
10: $\Delta f \leftarrow f_i - f_{i-1}$
11: $i \leftarrow i + 1$
12: **end while**
13: **return** f_{i-1}

on duty cycle r_{OD} and our optimizations aim at reducing P_{avg} or are constrained by an upper bound on P_{avg}. In this way, our adaptive search maintains duty cycle-awareness.

We estimate a design cone using a two-step procedure. First, given the frequency and voltage of a mode, we create LVT-only inverter chain and RVT-only NOR chain. The numbers of stages in the inverter and NOR chains are selected such that the delays of the chains match the reciprocal of the given frequency at the given voltage. Second, we simulate the inverter and NOR chains at different voltages to obtain the frequency versus voltage tradeoff curves that define the upper and lower boundaries of a design cone.

$3 + 1$ Problems

Recall from Section 3.1.2 that there are two types of $3 + 1$ problems. In the first type, given a nominal mode (f_{nom}, V_{nom}) and the frequency of another mode (f_{var}), we seek to find another voltage (V_{var}) that minimizes circuit power. We solve this problem with the following steps.

1. Find the design cone at the nominal mode (f_{nom}, V_{nom}).
2. Find the range of V_{var} defined by the intersections of f_{var} and boundaries of the design cone.
3. Apply the MIN_POWER procedure (with $f = f_{var}$) to obtain the desired V_{var}.

In the second type, given power constraints, a nominal mode (f_{nom}, V_{nom}) and a voltage (V_{var}), we seek to find the maximum frequency (f_{var}). We solve this problem with the following steps.

1. Find the design cone at the nominal mode (f_{nom}, V_{nom}).
2. Find the range of f_{var} defined by the intersections of V_{var} and boundaries of the design cone.
3. Apply the MAX_FREQ procedure (with $V = V_{var}$) to obtain the desired f_{var}.

The FIND_OD $2 + 2$ Problem

In the FIND_OD problem, only the nominal mode (f_{nom}, V_{nom}) and power constraints are given. Thus, we cannot apply the MAX_FREQ flow directly. Based on *Lemma 2*, we reduce the design space by searching for the overdrive mode within the design cone of the nominal mode. Further, we observe that a circuit implemented at a particular pair of nominal mode (e.g., mode ***A*** in Figure 3.9) and overdrive mode (e.g., mode ***B***) can also run at other overdrive modes (e.g., mode ***B'***) along its frequency versus voltage tradeoff curve (e.g., red dotted line). This implies that circuits implemented with a nominal mode (f_{nom}, V_{nom}) and any overdrive mode along one frequency versus voltage tradeoff curve will have similar circuit properties. The above observation reduces the number of MCMM circuit implementations during the adaptive search, in which we extract circuit properties for solutions in the design cone by generating a few trial circuits on different frequency versus voltage tradeoff curves. We solve the FIND_OD problem with the following steps.

1. Find the design cone of the nominal mode (f_{nom}, V_{nom}).
2. Find the intersections of the maximum supply voltage V_{max} and boundaries of the design cone. Define the minimum and maximum frequencies of these intersections as f_a and f_b, respectively.
3. Run MCMM SP&R with the given nominal mode and with overdrive modes defined by $\{f_a, f_b, (f_a + f_b)/2\}$ and V_{max}.
4. Extract circuit information.[23] Build or update the power model.

[23] Circuit information includes C_{load}, C_{gate}, r_{LVT}, P_{sw}, P_{int} and P_{leak}.

5. Estimate P_{avg}, based on the given r_{OD}, corresponding to feasible overdrive modes within the design cone. Find the maximum f_{OD} along with the corresponding V_{OD} (i.e., the overdrive mode) satisfying power constraints.

6. Run MCMM SP&R with the overdrive mode obtained in Step 5. Repeat Steps 4 to 6 until the difference in f_{OD} is less than a stopping criterion f_{stop} (e.g., f_{stop} = 10*MHz*).

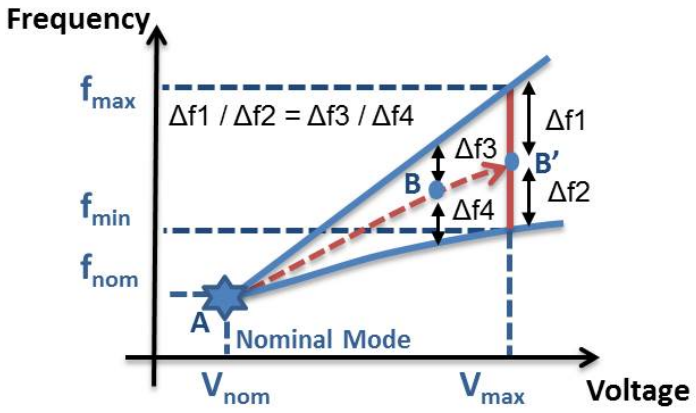


Figure 3.9: Projection of frequency and voltage pair at ***B*** to frequency at ***B'*** with predefined V_{max} for circuit property modeling.

The FIND_NOM $2 + 2$ Problem

The FIND_NOM problem is similar to the FIND_OD problem. We solve the FIND_NOM problem using the same methodology as for the FIND_OD problem, via the following steps.

1. Find the design cone of the overdrive mode (f_{OD}, V_{OD}).

2. Find the intersections of the minimum supply voltage V_{min} and boundaries of the design cone. Define the minimum and maximum frequencies of these intersections as f_a and f_b, respectively.

3. Run MCMM SP&R with the given overdrive mode and with nominal modes defined by $\{f_a, f_b, (f_a + f_b)/2\}$ and V_{min}.

4. Extract circuit information. Build or update the power model.

5. Estimate P_{avg}, based on the given r_{OD}, corresponding to feasible nominal modes within the design cone. Find the maximum f_{nom} along with the corresponding V_{nom} (i.e., the nominal mode) satisfying power constraints.

6. Run MCMM SP&R with the nominal mode obtained in Step 5. Repeat Steps 4 to 6 until the difference in f_{nom} is less than a stopping criterion f_{stop} (e.g., f_{stop} = 10*MHz*).

The FIND_VOLT $2+2$ Problem

Given f_{nom} and f_{OD}, we search for V_{nom} and V_{OD} to minimize P_{avg}. Finding the optimal V_{nom} and V_{OD} pair using exhaustive search incurs large runtime because there are many combinations of V_{nom} and V_{OD}. To reduce the runtime complexity, we propose an *approximate* optimization method – for each V_{nom}, we consider only one V_{OD}. From our studies, we observe that the *ratio of HVT cells to total cells* (λ) in the critical paths increases with the signoff voltage. This is because when the signoff voltage increases, paths become faster and more HVT cells are used in the critical paths to reduce power. As a result, for a fixed f_{nom}, $\lambda(V_{nom})$ increases with V_{nom}. Therefore, we heuristically select V_{OD} for a fixed V_{nom} based on the estimated $\lambda(V_{nom})$. More specifically, within a design cone, we define $\lambda(V_{nom})$ as

$$\lambda(V_{nom}) = (V_{LVT} - V_{OD})/(V_{LVT} - V_{HVT}) \tag{3.9}$$

where V_{LVT} and V_{HVT} are the minimum supply voltages at which the LVT and HVT inverter chains meet the f_{OD} requirement, and V_{OD} is the overdrive voltage. This is illustrated in Figure 3.10.

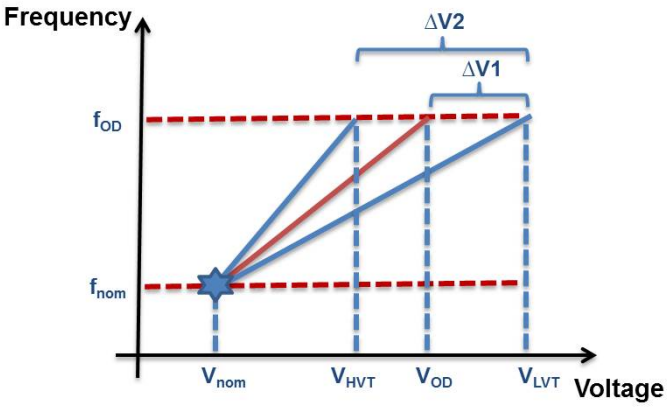


Figure 3.10: Illustration of $\lambda(V_{nom})$ calculation, where $\lambda(V_{nom}) = \Delta V1/\Delta V2$.

We denote the maximum supply voltage at the given technology node as V_{max}, and the minimum supply voltage at f_{nom} as V_{min}, which we assume can be determined by designers. To solve the FIND_VOLT problem, we first define two nominal modes, (f_{nom}, V_{min}) and (f_{nom},

V_{max}). We then determine the desired V_{OD} for each mode by solving a $3+1$ problem; from the V_{OD}, we calculate $\lambda(V_{max})$ and $\lambda(V_{min})$ using Equation (3.9). We then determine the desired $\lambda(V_{nom})$ for different nominal voltage values based on $\lambda(V_{min})$ and $\lambda(V_{max})$. Since $\lambda(V_{nom})$ increases with V_{nom}, we approximate $\lambda(V_{nom})$ as a linear function of V_{nom}.

$$\lambda(V_{nom}) = \frac{\lambda(V_{max}) - \lambda(V_{min})}{V_{max} - V_{min}} \times V_{nom} + \lambda(V_{min}) \tag{3.10}$$

Based on Equation (3.10), for each V_{nom} we determine a $\lambda(V_{nom})$ and the corresponding V_{OD}, as shown in Figure 3.11. Such an approximate optimization reduces the runtime complexity of the FIND_VOLT problem. Experimental results in Section 3.1.5 show that our approximate optimization can achieve similar results to the exhaustive search. Detailed steps to solve the FIND_VOLT problem are as follows.

1. Define two nominal modes (f_{nom}, V_{min}) and (f_{nom}, V_{max}). For each nominal mode, determine the V_{OD} with minimum P_{avg} by solving a $3+1$ problem.
2. Based on this V_{OD}, calculate $\lambda(V_{min})$ and $\lambda(V_{max})$.
3. Determine the relationship between V_{nom} and $\lambda(V_{nom})$ using Equation (3.10).
4. Run MCMM SP&R at $\{V_{min}, V_{max}, (V_{min} + V_{max})/2\}$ (with f_{nom}) and the corresponding V_{OD} (with f_{OD}) determined by λ values.
5. Extract circuit information. Build or update the power model.
6. Find V_{nom} and the corresponding V_{OD} that achieve minimum P_{avg} based on the power model.
7. Run MCMM SP&R with the V_{nom} and V_{OD} obtained in Step 6. Repeat Steps 5 to 7 until the difference in P_{avg} is less than a stopping criterion P_{stop} (e.g., P_{stop} = 2*mW*).

The FIND_FREQ $2+2$ Problem

Given V_{nom} and V_{OD}, we search for f_{nom} and f_{OD} to maximize f_{avg} (= $(1 - r_{OD}) \times f_{nom} + r_{OD} \times f_{OD}$). To reduce the runtime complexity of the problem, for each f_{nom}, we consider only one f_{OD}. We assume that the minimum (f_{min}) and maximum (f_{max}) frequencies can be empirically selected by designers or determined by the frequency requirements. To solve the

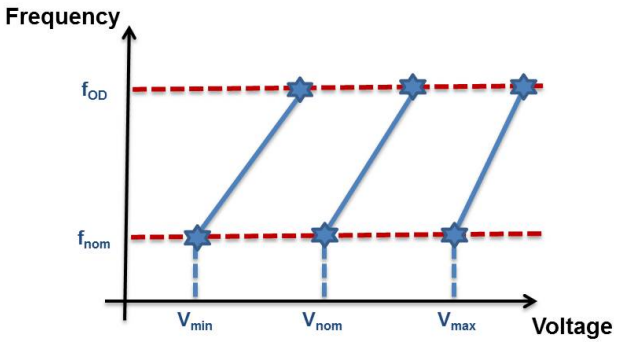


Figure 3.11: For each V_{nom}, we consider only one V_{OD}. The desired V_{OD} is determined based on $\lambda(V_{nom})$ and the design cone. $\lambda(V_{nom})$ is estimated as a linear function of V_{nom}. This approximation reduces runtime complexity but can achieve similar results to exhaustive search.

FIND_FREQ problem, we first calculate $\lambda(f_{min})$ and $\lambda(f_{max})$. We then approximate $\lambda(f_{nom})$ as a linear function of f_{nom}, i.e.,

$$\lambda(f_{nom}) = \frac{\lambda(f_{max}) - \lambda(f_{min})}{f_{max} - f_{min}} \times f_{nom} + \lambda(f_{min}) \tag{3.11}$$

where $\lambda(f_{nom})$ is the λ corresponding to f_{nom}. Detailed steps to solve the FIND_VOLT problem are as follows.

1. Use the given V_{nom} and empirically determined f_{min}, f_{max} to define two nominal modes (f_{min}, V_{nom}) and (f_{max}, V_{nom}). For each nominal mode, determine the f_{OD} with maximum f_{avg} by solving a $3 + 1$ problem.

2. Based on this f_{OD}, calculate $\lambda(f_{min})$ and $\lambda(f_{max})$.

3. Determine the relationship between f_{nom} and $\lambda(f_{nom})$ using Equation (3.11).

4. Run MCMM SP&R at $\{f_{min}, f_{max}, (f_{min} + f_{max})/2)\}$ (with V_{nom}) and the corresponding f_{OD} (with V_{OD}) determined by λ values.

5. Extract circuit information. Build or update the power model.

6. Find f_{nom} and the corresponding f_{OD} that achieve maximum f_{avg} under power constraints.

7. Run MCMM SP&R with the f_{nom} and f_{OD} obtained in Step 6. Repeat Steps 5 to 7 until the differences in f_{nom} and f_{OD} are less than a stopping criterion f_{stop} (e.g., f_{stop} = 10MHz).

3.1.5 Experimental Results

Our experiments use two designs – *AES* ($14K$ instances at $28nm$) and *JPEG* ($40K$ instances at $28nm$) – obtained from the *OpenCores* [243]. These designs are implemented using foundry $28nm$ RVT and LVT libraries. We characterize all libraries at operating voltages ranging from $0.80V$ to $1.10V$ (in steps of $20mV$) using *Synopsys SiliconSmart* [256]. The designs are synthesized using *Synopsys Design Compiler* [250] and then placed and routed using *Synopsys IC Compiler* [252]. We use *Synopsys PrimeTime* [254] for timing and power analyses, and *Synopsys HSPICE* [251] for all transistor-level modeling and simulations. We use *MATLAB* [239] to implement least squares regression and derive our power model.

In our experiments, we implement synthesis at both nominal and overdrive modes, and pick the mode which reports less power after routing. For each synthesized gate-level netlist, we run MCMM P&R with both nominal and overdrive modes. To eliminate tool noise, we execute each P&R run three times, perturbing the timing constraints by a small amount (i.e., 0.5% of the clock period) [93]. Unless otherwise specified, we use $r_{OD} = 50\%$ in our experiments. We run timing analysis at SS corner and power analysis at FF corner. All implemented designs have worst negative slacks (WNS) $\geq -30ps$.[24]

FIND_OD Problem

We study three instances of the FIND_OD problem. Table 3.2 shows the experimental setup, where P_{avg_max}, P_{peak_max} and V_{max} respectively constrain average power, peak power and signoff voltages.

Table 3.2: Experimental setup for the FIND_OD problem.

Case	Design	f_{nom} (*MHz*)	V_{nom} (V)	P_{avg_max} (*mW*)	P_{peak_max} (*mW*)	V_{max} (V)
1	*AES*	800	0.8	25	30	1.1
2	*AES*	800	0.8	35	40	1.1
3	*JPEG*	500	0.8	35	50	1.1

We implement four methods in our study of the FIND_OD problem. The *Signoff&Scale* method applies the traditional "signoff and scale" methodology, where we first sign off cir-

[24]The small WNS is due to the discrepancy between timing analysis in *Synopsys IC Compiler* [252] and in *Synopsys PrimeTime* [254].

cuits with the given nominal mode and then perform timing and power analyses with libraries characterized at higher voltages to search for the maximum overdrive frequency under power constraints. Note that we perform an additional MCMM P&R run to optimize power at both modes after the overdrive mode is selected. The *Proposed method* (in Section 3.1.4) searches for the overdrive mode using the proposed adaptive search within the design cone. The *Exhaustive search* explores the entire feasible design space for given design parameters. Specifically, we choose the signoff overdrive voltages within the range ($V_{SS\_sol} - 20mV$, $V_{SS\_sol} + 80mV$) with step sizes of $20mV$, and the overdrive frequencies within the range (f_{SS_sol} + 20*MHz*, f_{SS_sol} + 100*MHz*) with step sizes of 20*MHz*, where (f_{SS_sol}, V_{SS_sol}) is overdrive mode resulting from the *Signoff&Scale* method. We also compare to the method in [37], which searches for the overdrive mode within the design cone of the nominal mode. Table 3.3 summarizes our experimental results.

Table 3.3: Metrics of circuits for the FIND_OD problem.

	AES (Case 1)				*AES* (Case 2)				*JPEG* (Case 3)			
	Signoff &Scale	*Proposed method*	*Exhaustive search*	Method in [37]	*Signoff &Scale*	*Proposed method*	*Exhaustive search*	Method in [37]	*Signoff &Scale*	*Proposed method*	*Exhaustive search*	Method in [37]
f_{OD} (*MHz*)	1220	1270	1280	1260	1400	1470	1480	1440	800	845	880	820
V_{OD} (V)	0.98	0.96	0.98	1.02	1.06	1.02	1.06	1.10	1.00	0.98	0.98	1.02
Area (μm^2)	11591	12229	12051	11474	11561	12527	11991	11457	55125	57225	54549	53207
#Cells	13495	13919	13781	13240	13454	14163	13753	13393	43309	45029	41456	41518
$\%LVT$	81	89	87	75	81	90	87	77	45	41	41	46
P_{avg} (*mW*)	19.4	20.7	20.9	21.2	24.5	25.5	26.6	26.5	34.0	35.1	34.5	35.3
P_{peak} (*mW*)	27.3	28.9	29.7	30.9	37.4	38.5	41.0	41.5	48.6	50.1	49.1	51.5
#P&R runs	2	4	30	10	2	4	30	8	2	4	30	10

The results show that the *Proposed method* achieves up to 6% improvement in overdrive performance compared to the *Signoff&Scale* method while maintaining similar area and power. This is a significant improvement, considering that even 20% improvement in performance per new technology generation is now quite difficult to achieve. The results also show that the overdrive frequency obtained from the *Proposed method* is within 4% of that obtained from the *Exhaustive search*, while the *Proposed method* uses less than 14% of the *Exhaustive search* runtime. Moreover, the *Proposed method* achieves similar results compared to the method in [37] for the smaller design (i.e., *AES*), and 3% improvement in performance for the larger design (i.e., *JPEG*). We also note that the number of SP&R runs required by the method in [37] can increase significantly with a large performance range. By contrast, our *Proposed method* is more scalable due to its use of adaptive search, which can estimate the optimal overdrive mode and is able to converge to a near-optimal solution after a small number of SP&R runs.

FIND_VOLT Problem

We study two instances of the FIND_VOLT problem (Table 3.4), where V_{max} is the maximum signoff voltage.

Table 3.4: Experimental setup for the FIND_VOLT problem.

Case	Design	f_{nom} (*MHz*)	f_{OD} (*MHz*)	V_{max} (V)
4	*AES*	1000	1300	1.1
5	*JPEG*	600	800	1.1

Table 3.5 shows results for the FIND_VOLT problem achieved by the *Proposed method* (in Section 3.1.4), *Exhaustive search* and the method in [37]. The *Exhaustive search* searches the nominal voltage within the range $(0.80V, 0.98V)$ with step sizes of $20mV$. For each V_{nom}, we search for V_{OD} within the range ($1.05 \times V_{nom}$, $1.2 \times V_{nom}$) with step sizes of $20mV$.

Table 3.5: Metrics of circuits for the FIND_VOLT problem.

	AES (Case 4)			*AES* (Case 5)		
	Proposed method	*Exhaustive search*	Method in [37]	*Proposed method*	*Exhaustive search*	Method in [37]
V_{nom} (V)	0.88	0.92	0.96	0.82	0.82	0.82
V_{OD} (V)	0.98	1.06	1.08	0.92	0.92	0.92
Area (μm^2)	12084	12439	10150	55276	55276	55276
#Cells	13911	14124	12276	45469	45469	45469
$\%LVT$	87	71	68	49	49	49
P_{avg} (*mW*)	24.5	23.9	24.7	32.4	32.4	32.4
P_{peak} (*mW*)	30.5	30.7	31.1	40.5	40.5	40.5
#P&R runs	7	42	11	7	42	15

Results in Table 3.5 show that the *Proposed method* achieves less than 3% power overhead and 6× runtime reduction compared to the *Exhaustive search*. We also observe that the *Proposed method* achieves less average power and runtime compared to the method in [37].

Duty Cycle-Awareness Validation

Our methodology is duty cycle-aware. We optimize design *AES* (under the context of the FIND_OD problem) with different duty cycles (i.e., $r_{OD} = 0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9$) and compare metrics of the implemented circuits. In the experiments, we assume upper bounds of average power and signoff voltages as $30mW$ and $1.1V$, respectively. The nominal mode is

(800*MHz*, 0.8V). Further, we run frequency scaling on the implemented circuits to evaluate their maximum f_{OD} with different duty cycles. More specifically, with an assumed duty cycle, we increase f_{OD} with step sizes of 5*MHz*, and for each f_{OD} we choose the minimum V_{OD} without timing violation for power analysis. We keep increasing the f_{OD} until P_{avg} reaches its upper bound.

Table 3.6: Metrics of circuits implemented with different duty cycles (r_{OD_opt}). r_{OD_eva} is the duty cycle for evaluation.

r_{OD_opt}	f_{OD} (*MHz*)	V_{OD} (V)	Area (μm^2)	#Cells	%LVT	f_{max} (*MHz*) with r_{OD_eva} =				
						0.1	0.3	0.5	0.7	0.9
0.1	1720	1.10	12553	14233	90	**1660**	1410	1225	1110	1035
0.3	1440	1.04	11966	13725	87	1615	**1465**	1245	1130	1070
0.5	1220	0.96	11995	13719	87	1600	1445	1230	1125	1070
0.7	1110	0.92	11987	13642	86	1600	1450	1235	**1130**	1075
0.9	1050	0.90	11947	13682	85	1600	1445	1225	1125	**1075**

Table 3.6 shows metrics of circuits implemented with different duty cycles. We observe that f_{OD} and V_{OD} reduce with a larger r_{OD_opt}. In other words, optimization (or signoff) with a small r_{OD_opt} results in a fast design. This is because given particular nominal and overdrive modes, P_{avg} increases with r_{OD_opt}, and power constraints limit the increase of f_{nom} (and V_{OD}) during the optimization. Results in Table 3.6 also show pessimism of inaccurate prediction for r_{OD}. For example, if the actual r_{OD} is 0.1 but the optimization assumes $r_{OD} = 0.9$, the performance penalty will be 4%.

Another observation is that the circuit optimized with a particular r_{OD} usually achieves the maximum f_{OD} when evaluated with the corresponding r_{OD} (Table 3.6, values in bold), as compared to circuits optimized with other values of r_{OD}. This again confirms the duty cycle-awareness of the proposed flow.

3.1.6 Conclusions

We study the multi-mode signoff optimization problem and introduce the concept of equivalent dominance among signoff modes. We show that for a multi-mode design, the modes for signoff must maintain a mutual equivalent dominance condition to avoid overdesign. Based on the properties of equivalent dominance, we propose guidelines and efficient methodologies to search for the optimal modes for overdrive signoff. The proposed methodologies are duty cycle-aware and can successfully determine the signoff modes that reduce lifetime energy (i.e., P_{avg}). Our experimental results indicate that the proposed methodologies can identify signoff

modes which lead to up to 6% performance improvement compared to the traditional "signoff and scale" methodology. Our experiments further show that circuits signed off with our proposed flow have less than 3% overhead in average power compared to the essentially optimal results obtained through exhaustive search but with $6\times$ runtime reduction. Moreover, our proposed methodology achieves up to 3% performance improvement and less average power compared to the previous work in [37].

Our ongoing work seeks (i) consideration of additional tradeoffs of design metrics such as circuit area, reliability and design time; (ii) more accurate estimation of the design cone in advanced technology nodes, in particular, considering impacts of increased wire resistance; (iii) consideration of process corners and temperature in the approximation of design cone; and (iv) efficient methodologies for multi-mode signoff with more than two modes.

3.2 On Aging-Aware Signoff for Circuits with Adaptive Voltage Scaling

Transistor aging due to bias temperature instability (BTI) is a major reliability concern in sub-$32nm$ technology. To compensate for aging, designs now typically apply adaptive voltage scaling (AVS) to mitigate performance degradation by elevating supply voltage. Since varying the supply voltage also causes the BTI degradation to vary over lifetime, this presents a new challenge for margin reduction in the context of conventional signoff methodology, which characterizes timing libraries based on transistor models with precalculated BTI degradations for a given IC lifetime. In this section, we study the conditions under which a circuit with AVS requires additional timing margin during signoff. Then, we propose two heuristics for chip designers to characterize an aging-derated standard-cell timing library that accounts for the impact of AVS during signoff. According to our experimental results, this aging-aware signoff approach avoids both overestimation and underestimation of aging – either of which results in power or area penalty – in AVS-enabled systems. Further, we compare circuits implemented with the aging-aware signoff method based on aging-derated libraries versus those based on a flat timing margin. We demonstrate that the flat timing margin method is more pessimistic, and that the pessimism can be mitigated by AVS.

To ensure that circuits can meet frequency requirements at different operating conditions, designers must *sign off* circuits by verifying timing correctness with timing libraries characterized at specific voltages and process corners. As technology nodes advance, BTI is a major aging mechanism, particularly in sub-$32nm$ CMOS technology. The BTI effect increases the

threshold voltage ($|V_{th}|$) of a MOS transistor, resulting in a time-dependent timing degradation in VLSI circuits [90] [100]. It is mandatory to consider the BTI effect in modern timing signoff recipes – via 10-year timing libraries, flat V_{dd} margin, etc. – to ensure that circuits will operate correctly over their entire lifetimes.

AVS is a design technique that compensates for BTI-induced circuit performance degradation by increasing the supply voltage (V_{dd}) of a circuit [11] [119] [217]. Since supply voltage is increased to compensate for BTI-induced timing degradation, the supply voltage of the circuit at the end of lifetime (V_{final}) is higher than the supply voltage at the beginning of lifetime (V_{init}). As illustrated in Figure 3.12, a higher V_{init} leads to a larger V_{final} because the higher V_{init} causes a larger BTI-induced timing degradation, which in turn requires higher supply voltages to compensate for the timing degradation. Therefore, when V_{init} is sufficiently large, the V_{final} will be clamped to the maximum allowed voltage (V_{max}).[25] We define $V_{critical}$ as the minimum V_{init} with V_{final} = V_{max}. Since V_{final} cannot exceed V_{max}, signoff margin for aging is required when $V_{init} \geq V_{critical}$.

We address two central questions. First, what determines $V_{critical}$, which determines whether additional margin is required for signoff? Second, what is the best practice for AVS- and aging-aware signoff when $V_{init} \geq V_{critical}$? Existing signoff methods to account for aging include (i) applying a flat timing margin (henceforth, *flat margin*) in signoff and (ii) characterizing aging-derated timing libraries (henceforth, *derated libraries*) to model device-specific aging effects. Method (i) requires only a minimal change in the existing signoff flow, but applying a timing margin for the entire circuit may incur large area and power penalties. On the other hand, it is difficult to characterize the derated library in Method (ii) because BTI degradation is worse when V_{dd} is higher but circuit delay is larger when V_{dd} is lower. If the derated library is optimistic, the estimated circuit delay during signoff is less than the actual delay during operation. This will lead to a higher V_{dd} and power consumption than designers anticipate at signoff. If the derated library is pessimistic, the estimated circuit delay during signoff is larger than the actual delay at runtime. As a result, circuit area will unnecessarily increase because larger cell sizes are required to meet the timing constraints. With this in mind, we also study the design overheads when derated libraries are not properly characterized, as well as the guidelines to define BTI- and AVS-aware signoff corners that guarantee timing correctness with little design overheads.

There have been many studies on the optimization of V_{dd} in AVS to mitigate BTI-degradation while minimizing circuit power [11] [48] [119] [120] [129] [150] [186]. These

[25] The maximum allowed voltage can be limited by many factors such as electromigration, system requirements, etc. The black dotted line is unachievable due to this V_{max} limitation.

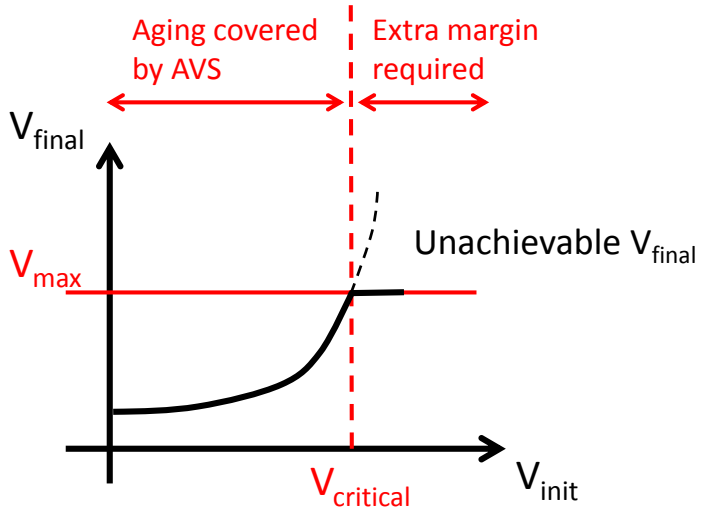


Figure 3.12: Difference between V_{init} and V_{max} reduces as V_{init} approaches $V_{critical}$.

previous works focus on the application of AVS to mitigate BTI aging, but none of them study the AVS- and aging-aware signoff questions mentioned above. The previous works assume that a circuit is designed and signed off with timing libraries without BTI effect. As shown in Figure 3.12, such an assumption fails when V_{init} exceeds $V_{critical}$. Although a BTI-aware timing analysis can be applied after signoff [120], this requires multiple iterations of signoff and resizing or other *engineering change orders* (ECOs) before the circuit implementation converges. Resolving this inconsistency is one of the subjects of our present investigation. Our contributions are as follows.

- We analyze the factors that determine $V_{critical}$, which can help circuit designers to decide whether additional signoff margin is required.
- We sign off benchmark circuits using different derated libraries and compare metrics (e.g., area and power) of the resulting circuit implementations. Our experimental results show that circuits signed off using different derated libraries have up to 38% area or 21% dynamic power overheads for the same frequency requirements.
- We analyze the impact of BTI degradation and the inconsistency of voltages used for characterizing libraries and aging, respectively, and propose selection guidelines for the voltages that characterize the aging effect in a circuit with AVS. We conduct experiments to verify our methodologies with a foundry $28nm$ fully-depleted silicon-on-insulator (FD-SOI) technology.

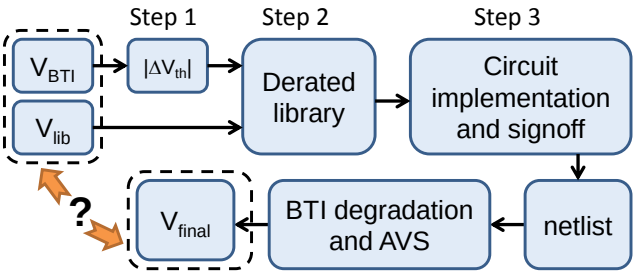


Figure 3.13: The upper part of this figure illustrates a signoff flow using a derated library. The lower part of this figure illustrates that AVS increases the voltage of the circuit to compensate for BTI degradation.

- We study different aging-aware signoff methodologies by comparing circuits implemented using a flat margin and with those implemented using derated libraries. We conclude that the flat margin method is simpler but more conservative than the derated library method. We also demonstrate that this pessimism can be mitigated by AVS.

3.2.1 Aging-Aware Signoff

Figure 3.13 illustrates the interactions among library characterization, circuit signoff and AVS. Steps 1 to 3 in the upper part of the figure show a typical signoff flow including the characterization of a derated library. The three steps are described as follows.

1. In Step 1, the magnitude of BTI degradation ($|\Delta V_{th}|$) is estimated using an aging model. Note that the voltage applied in the aging model, which we denote by V_{BTI} (V_{BTI} is used to calculate the $|\Delta V_{th}|$ for derated library characterization), significantly influences the $|\Delta V_{th}|$ that results from BTI degradation [197]. Therefore, the selection of V_{BTI} affects the derated library.

2. In Step 2, the extracted $|\Delta V_{th}|$ is used in transistor models to characterize a derated library that accounts for BTI degradation. During the library characterization, transistors and standard cells are simulated at a possibly different voltage level, which we denote by V_{lib}.

3. In Step 3, with the derated library, circuit designers can implement and sign off a circuit.

During runtime (lower part of Figure 3.13), AVS increases the V_{dd} of the circuit to compensate for BTI degradation. This will lead to a higher V_{dd} at the end of circuit lifetime (V_{final}). Note that V_{lib}, V_{BTI} and V_{final} could be different from each other. For instance, V_{final} is a result

of AVS to compensate for BTI degradation which varies depending on circuit implementation. Also, guardbanding for the worst-case operating condition during library characterization will lead to different V_{lib} and V_{BTI}. This is because the worst-case BTI degradation happens when V_{BTI} is high but the worst-case gate delays happen when V_{lib} is low. Moreover, circuit designers do not know V_{final} before the circuit is implemented.

Signoff with Derated Library

In a typical timing signoff methodology, meeting timing constraints with predefined corner libraries implies that the circuit will work correctly at the target specification. This is because the corner libraries are characterized at worst-case operating conditions. Thus, to characterize a BTI-derated library for signoff, traditional methodology considers the worst-case transistor degradation due to the BTI effect. Our present work focuses on library characterization for signoff of setup-time checks, since the main effect of BTI aging is to increase delay in data paths.

Characterization of a derated library is commonly performed in two steps. First, transistor aging is estimated at a worst-case scenario defined by the total time of BTI stress, the temperature, and the voltage (V_{BTI}) being applied to the transistors. Note that this BTI degradation estimation is pessimistic for an AVS circuit because V_{BTI} is defined as a constant for the entire lifetime, whereas the voltage of an AVS circuit is initially smaller and gradually increases during circuit lifetime. Second, the transistor aging ($\Delta|V_{th}|$) calculated from the first step is included in transistor models for library characterization. During derated library characterization, we must also fix the operating voltage (V_{lib}) of the transistors and standard cells. The values of V_{BTI} and V_{lib} could be different because the worst-case corner for V_{BTI} is at the maximum allowed voltage (higher voltage increases $\Delta|V_{th}|$), while the worst-case corner for V_{lib} is at the minimum allowed voltage (lower voltage increases gate delay).

Worst-Case BTI Degradation

Note that the BTI-induced timing degradation is affected by the total stress time (i.e., total time when transistors are on), which varies depending on circuit activity. The actual circuit activity is very difficult to capture because it is determined by circuit usage. Since it is impractical for any known AVS monitor to capture the detailed circuit activity of each transistor in a circuit, we assume that designers must consider a worst-case scenario at signoff.

Velamala et al. in [199] show that worst-case timing degradation occurs when critical paths experience a long *DC BTI stress* (i.e., transistors are always under BTI stress). However,

assuming a DC BTI stress may be too pessimistic: a typical CMOS circuit usually switches during operation, and exhibits an *AC BTI stress* (i.e., transistors experience alternate BTI stress and recovery phases). The measurement results in [82] and [90] show that the amount of BTI degradation is not sensitive to *stress duty cycle* (i.e., the ratio of total stress time to total operating time) when the duty cycle ranges from 20% to 80%. This means that we can approximate the BTI degradation in a typical CMOS circuit by assuming an AC BTI stress with 50% duty cycle. In the studies reported below, we consider both DC and AC aging scenarios with $125°C$ operating temperature.[26]

Adaptive Voltage Scaling (AVS)

To study BTI degradation of a circuit with AVS, we assume that the circuit monitors its maximum frequency (f_{max}) in a discrete-time manner. Whenever the f_{max} of the circuit is lower than a predefined target frequency (f_{target}), the V_{dd} will be increased by a V_{step} (where V_{step} is an attribute of the voltage regulator). After the V_{dd} adjustment, the AVS circuitry will evaluate f_{max} and continue to increase V_{dd} until $f_{max} \geq f_{target}$. The AVS mechanism is illustrated in Figure 3.14. In our discussion, we use t to denote time, Δt to denote the time interval between

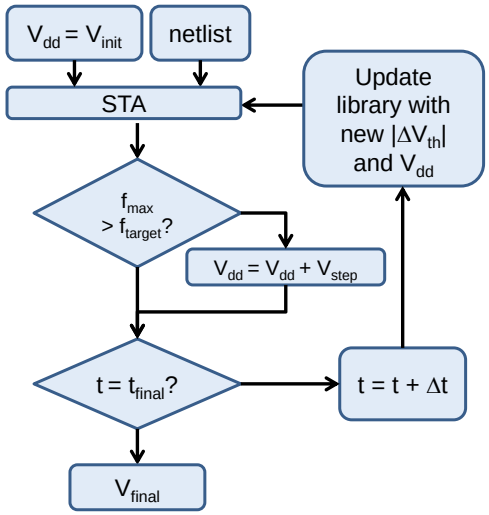


Figure 3.14: Experimental flow to emulate AVS mechanism.

successive AVS calibrations, t_0 to denote the initial time when the circuit starts to operate, and t_{final} to denote the end of circuit lifetime. The V_{dd} of the circuit at the beginning of its lifetime (i.e., the minimum voltage needed to meet the frequency requirement at t_0) is denoted by V_{init}.

[26] Although temperature profile is spatially nonuniform across a chip, we use the highest operating temperature ($125°C$) in our analysis to estimate the worst-case BTI degradation.

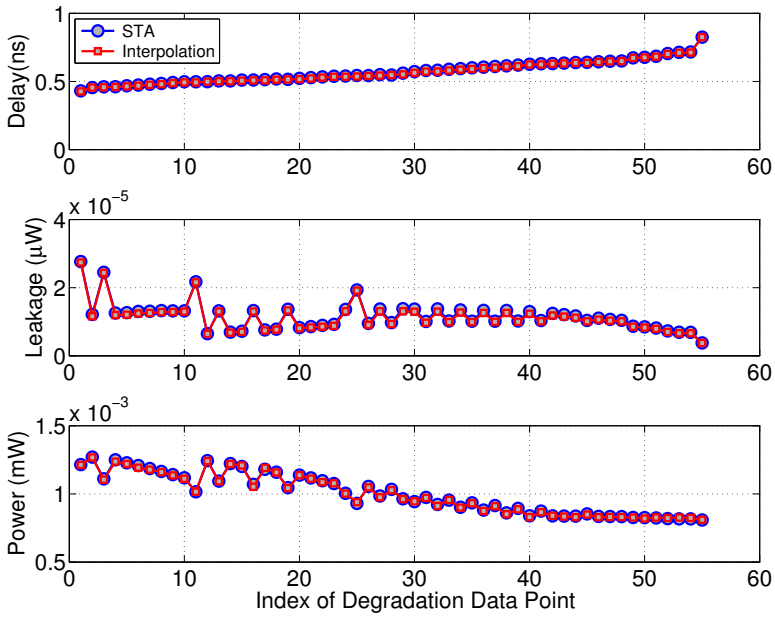


Figure 3.15: The average errors between the actual and the interpolated delay, leakage power, and dynamic power values at sampled points are 0.80%, 3.50%, and 0.57%, respectively.

The *update library* step in Figure 3.14 is very slow if we characterize a library whenever V_{lib} or $\Delta|V_{th}|$ is changed. To speed up the simulation runtime, we precharacterize a set of libraries with different V_{lib} and $\Delta|V_{th}|$. To obtain the f_{max} of a circuit at specific V_{lib} and $\Delta|V_{th}|$, we simulate the circuit with all the precharacterized libraries and estimate the f_{max} value by interpolation with spline polynomial functions. Circuit leakage power and dynamic power are estimated similarly. The lifetime leakage power and dynamic power are obtained by averaging over all timesteps. Figure 3.15 shows that the delay, leakage power and dynamic power estimations obtained from the interpolation have only 0.80%, 3.50%, and 0.57% maximum error, respectively, compared to values obtained by characterizing libraries at the sampled points.[27] All experiments are based on a commercial (i.e., production PDK with complete EDA tool enablement) foundry $28nm$ FDSOI technology.

3.2.2 Guidelines for Characterization of Derated Libraries

To study the relationship between V_{BTI} and V_{final}, we implement a given circuit using a library characterized at the nominal voltage (V_{nom}) of the process technology (V_{lib} = V_{nom}),

[27]The data points in Figure 3.15 are sorted with respect to delay values. Thus, the leakage and/or power plots can be nonmonotonic.

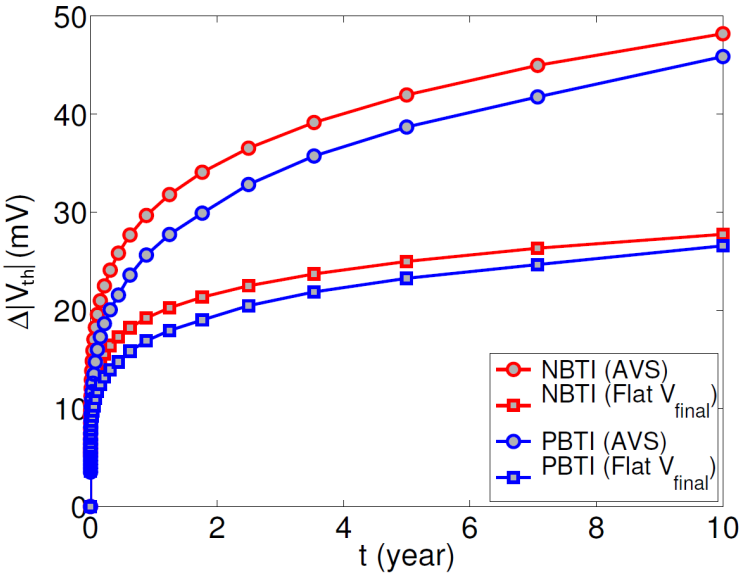


Figure 3.16: $|\Delta V_{th}|$ of PBTI and NBTI of a circuit (*MPEG2*) with a flat $V_{BTI} = V_{final}$, and with AVS, over circuit lifetime. The results show that the difference between a flat V_{dd} and AVS is less than $10mV$, and that this difference becomes smaller toward the end of circuit lifetime.

with the assumption that there is no BTI degradation. We then use the flow in Figure 3.14 to obtain the V_{final} of the circuit (lifetime = 10 years, DC BTI degradation). Figure 3.16 shows the $\Delta|V_{th}|$ with AVS compared to the case where V_{final} is applied to the same circuit throughout circuit lifetime. During the early lifetime, the BTI degradation ($\Delta|V_{th}|$) for the adaptive V_{dd} case (AVS) is less than that for the fixed V_{final} case. This is because the adaptive V_{dd} case has a smaller V_{dd} value at early lifetime, and BTI degradation increases with V_{dd}. However, due to the front-loaded nature of BTI degradation [39], ΔV_{th} difference between the fixed V_{final} and the AVS cases quickly diminishes.

The simulation results in Figure 3.16 show that we can estimate the degradation of an AVS circuit by assuming a constant V_{final} throughout circuit lifetime. This approximation slightly overestimates the $\Delta|V_{th}|$, but the overestimation is very small. In other words, we can characterize a derated library using V_{final} for signoff (i.e., $V_{BTI} = V_{final}$).

Note that the assumption of a constant V_{final} throughout circuit lifetime implies that $V_{lib} = V_{final} = V_{BTI}$. To understand what is the appropriate setup for V_{lib}, we analyze the implications when $V_{lib} \neq V_{BTI}$. When $V_{lib} > V_{BTI}$, the library characterization is optimistic because we assume that the operating voltage is higher than the voltage that defines BTI degradation. This violates the principle of having a derated library that defines the worst-case condition. Thus, we

should not use a V_{lib} that is greater than the V_{BTI}. On the other hand, having $V_{lib} < V_{BTI}$ means that the library characterization is pessimistic. However, there is no reason to be more pessimistic, because the degradation obtained from V_{BTI} is already slightly pessimistic. We conclude that having $V_{lib} = V_{final}$ is a reasonable option to avoid being optimistic or overly pessimistic in library characterization.

Of course, the main obstacle to library characterization with $V_{lib} = V_{BTI} = V_{final}$ is that this requires knowledge of the V_{final} of an AVS circuit, which is not available in the early design stages when the actual circuit is not fully implemented. Indeed, to obtain the V_{final}, we need to implement a circuit with a library, which requires V_{lib} and V_{BTI}. To overcome this "chicken and egg" problem, we analyze how circuit delay varies when subjected to changes in $|V_{th}|$ and V_{dd}. In the following, Equation (3.12a) is from [199].

$$\frac{\Delta d^{path}}{d^{path}} = \frac{\Delta V_{dd}}{V_{dd}} - \frac{\Delta V_{dd} - |\Delta V_{th}|}{V_{dd} - |V_{th0}|} \tag{3.12a}$$

$$= \frac{-|V_{th0}|}{V_{dd} \cdot (V_{dd} - |V_{th0}|)} \cdot \Delta V_{dd} + \frac{1}{V_{dd} - V_{th0}} \cdot |\Delta V_{th}| \tag{3.12b}$$

$$\frac{\Delta d^{path}}{d^{path}} = \frac{-|V_{th0}| \cdot b_{V_{dd}} \cdot \Delta V_{dd}}{V_{dd} \cdot (V_{dd} - |V_{th0}|)} + \frac{b_{V_{th}} \cdot |\Delta V_{th}|}{V_{dd} - |V_{th0}|} \tag{3.12c}$$

$$r_b = \frac{b_{V_{th}}}{b_{V_{dd}}} \tag{3.12d}$$

We use d^{path} to denote nominal path delay, and Δd^{path} to denote change in path delay due to ΔV_{dd} and $\Delta|V_{th}|$. $|V_{th0}|$ is the value of $|V_{th}|$ at time t_0 (i.e., when the circuit is fresh). In Equation (3.12c), we introduce parameters $b_{V_{dd}}$ and $b_{V_{th}}$ to represent sensitivities of a path delay (or a cell delay) to V_{dd} and $|V_{th}|$. In this analysis, we simulate a path (or a cell) with 153 $\{V_{dd}, V_{thn}, V_{thp}\}$ combinations using *Synopsys HSPICE* [251] and then apply linear regression (based on Equation (3.12c)) to extract $b_{V_{dd}}$ and $b_{V_{th}}$ for the corresponding path (or cell). This result is based on the foundry $28nm$ FDSOI NVT device model. The ratio of $b_{V_{th}}$ to $b_{V_{dd}}$ (i.e., r_b) indicates whether the path (or cell) is more sensitive to V_{dd} elevation or aging. Further, we emulate the AVS mechanism as explained in Figure 3.14. We assume V_{init} = $0.90V$, 10 years DC BTI stress, and a targeted path (or cell) delay equal to 101% of the path (or cell) delay at t_0.[28] After the AVS emulation, we calculate the $V_{final} - V_{init}$ after 10 years of DC BTI stress. The results in Table 3.7 imply the following.

[28] We use Equation (3.12c) and SPICE (instead of the STA tool) to estimate delay.

1. When the cell chain is composed of a set of diverse cells (Row 13 in Table 3.7)[29] the r_b of the cell chain converges to a value similar to that of chains composed of single-type cells (i.e., 0.55 versus 0.53, 0.51, 0.62, 0.53, 0.56 from AND2, OR2, NOR2, NAND2 and XOR2 chains, respectively.)

2. The value of $V_{final} - V_{init}$ shows a similar trend as the r_b, i.e., the $V_{final} - V_{init}$ of a chain of diverse cells is similar compared to single-type cell chains.

3. From Rows 11 and 12 in Table 3.7, the cell ordering in a path has negligible effect on r_b and $V_{final} - V_{init}$.

Since a setup timing-critical path typically passes through many different cells, $V_{final} - V_{init}$ of setup timing-critical paths will tend to converge to a value (cf. the law of large numbers). This observation lies at the root of the success in practice of our heuristic, which estimates V_{final} by averaging the V_{final} of different cell chains.

Table 3.7: Result of AVS emulation with different chain lengths, cell types, and cell type orderings using SPICE.

	Cell type		$V_{final} - V_{init}(mV)$	r_b
1	AND2	single cell	10	0.44
2		chain	14	0.53
3	OR2	single cell	7	0.39
4		chain	13	0.51
5	NOR2	single cell	28	0.96
6		chain	17	0.62
7	NAND2	single cell	29	1.00
8		chain	13	0.53
9	XOR2	single cell	20	0.73
10		chain	15	0.56
11	Mix of 5 cells (order 1)	-	17	0.63
12	Mix of 5 cells (order 2)	-	16	0.61
13	Mix of 14 cells	-	14	0.55

Results in Figure 3.17 show the V_{final} of different benchmark designs and standard cell chains. One subtle factor that affects V_{final} is the *delay margin* of the circuit. Delay margin (denoted by α) is defined as the difference (normalized to the signed-off circuit delay) between

[29]This set includes AND2, OR2, NOR2, NAND2, XOR2, inverters, and buffers.

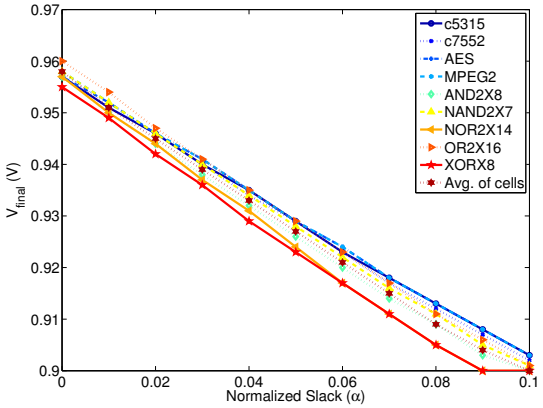


Figure 3.17: The relationship between V_{final} and α for different cells. α is the delay margin at signoff. The curves vary with different gate complexity and topology. The degradation is assumed to be with DC stress.

the target delay and the delay of the signed-off circuit at t_0 (denoted by $d_{t=0}^{chip}$). That is,

$$\alpha = \frac{d_{target}^{chip} - d_{t=0}^{chip}}{d_{target}^{chip}}, \qquad d_{target}^{chip} = \frac{1}{f_{target}} \tag{3.13}$$

Figure 3.17 shows that the V_{final} values are within a range of $< 10mV$ across all designs for α ranging from 0 to 0.1. This observation agrees with our analysis in Table 3.7 that we do not need design-specific analysis to obtain the relationship between V_{final} and α.

To estimate the V_{final} versus α curve of a circuit (before the circuit is implemented), we assume that the critical path of the circuit is composed of a mix of different cell types. Thus, we model the V_{final} versus α curve by averaging the curves from various cell types. We choose gates from the following categories to increase the gate diversity: (1) inverting and non-inverting gates, (2) PMOS-dominated gates, and (3) NMOS-dominated gates. Our simulation results in Figure 3.17 show that the maximum error of (V_{final}) among different circuits and cell chains is about one V_{step} ($10mV$) for different α.

In summary, we can characterize a derated library for an AVS circuit if the following AVS-related information is available: V_{init}, V_{step}, Δt and f_{target} (relative to circuit f_{max} at t_0).

3.2.3 Experimental Results for Signoff with Derated Libraries

Aging Model

To predict the impact of BTI on design performance, we use the analytic model from [197]. The $|V_{th}|$ degradation of a MOS transistor is given as

$$\begin{aligned} |\Delta V_{th}| &= \sqrt{{K_v}^2 \cdot (t - t_0)^{\frac{1}{n}}} \\ K_v &= A \cdot t_{ox} \cdot \sqrt{C_{ox}(V_{gs} - V_{th})} \cdot [1 - \frac{V_{ds}}{\beta(V_{gs} - V_{th})}] \\ &\quad \times exp(\frac{V_{gs}}{E_o t_{ox}}) \cdot exp(\frac{-Ea}{kT}) \end{aligned} \tag{3.14}$$

where t is the total stress time of a transistor, t_0 is the time when a circuit is turned on for the first time, k is the Boltzmann constant, t_{ox} is transistor oxide thickness, T is temperature, V_{gs} is gate-to-source voltage, and V_{ds} is drain-to-source voltage. We assume that both V_{gs} and V_{ds} are the same as V_{BTI}. β, n and A are fitting parameters with values as listed in Table 3.8.[30]

To explore circuit-level performance degradation, we use the aforementioned calibrated transistor degradation model along with the foundry $28nm$ FDSOI library and the SPICE model in its PDK. The model includes both LVT cells and NVT cells.

We obtain timing and power of the circuits using *Synopsys PrimeTime* [254]. To model BTI degradation with varying V_{dd}, we use the technique in [11] [199].[31]

Circuit Implementation

To evaluate the impact of AVS on aging-aware signoff, we compare the area and power of circuits that are signed off with different derated libraries. We set up experiments by implementing four benchmark circuits: *C5315*, *C7552* [24], *AES*, and *MPEG2* [243]. We use *Synopsys SiliconSmart* [256] to characterize libraries based on the worst-case corner of the $28nm$ FDSOI SPICE model for both LVT and NVT cells. The circuits are obtained through the following steps:

[30] We fit the parameters A, E_0, and β based on a set of BTI data in [215]. Then, we extract the values of n for PBTI and NBTI from their corresponding measurement plots in [215]. The value of E_a is obtained from [197].

[31] This technique can be summarized as follows. Whenever V_{dd} is changed at time t_i, we record the accumulated $\Delta|V_{th}|$ as $\Delta V_{th_i}^{acc}$. Based on the $\Delta V_{th_i}^{acc}$, we calculate the *effective stress time* t_i' using the relationship between ΔV_{th} and t, which can be obtained from the aging model (3.14) with $V_{ds} = V_{gs} = V_{dd} + V_{step}$. After that, the $\Delta|V_{th}|$ for the i^{th} time interval ($\Delta|V_{th_i}|$) can be obtained by calculating the difference between $\Delta|V_{th}|$ at t_i' and at $t_i' + \Delta t$. Finally, the accumulated $|V_{th}|$ degradation is given as

$|\Delta V_{th_i + \Delta t}^{acc}| = (|\Delta V_{th_i}^{acc}|^{\frac{1}{n}} + |\Delta V_{th_i}|^{\frac{1}{n}})^n$.

Table 3.8: Parameters of PBTI and NBTI aging models.

	PBTI	NBTI
n	3.3	2.5
A	$4.52e^{-3}$	
β	0.85	
$E_0(MV/cm)$	0.15	
$E_a(eV)$	0.13	
$t_{ox}(nm)$	1.15	1.20
V_{th} (V)	0.494	0.492

Table 3.9: Reference voltages used in our experiments.

	Voltage (V)	
	$28nm$ NVT	$28nm$ LVT
V_{max}	1.10	1.10
V_{init}	0.9	0.9
V_{heur1} (DC)	0.97	0.97
V_{heur2} (DC)	0.94	0.94
V_{heur1} (AC)	0.94	0.94
V_{heur2} (AC)	0.92	0.92

Table 3.10: Clock constraints for the power-area tradeoff experiments.

	Clock constraint (GHz)	
	$28nm$ NVT	$28nm$ LVT
C5315	1.82	2.22
C7552	1.82	2.00
AES	0.91	1.14
MPEG2	0.98	1.30

1. Define V_{init} = $0.9V$, Δt = 3 days, $V_{step} = 0.01V$ and f_{target} for each benchmark circuit. The clock constraints of the four designs are listed in Table 3.10.

2. Implement each circuit using a library characterized with V_{lib} = $0.9V$, $\Delta|V_{th}|$ = 0.

3. Mitigate EDA tool "noise" by making three separate synthesis, place and route runs for each benchmark circuit with $\{$-1, +0, +1$\}ps$ perturbation of the clock constraint with each run generating a circuit [93]. Then, report metrics for the circuit with minimum power among the three candidate circuits thus produced.

4. Run the flow in Figure 3.14 to ensure that the circuit does not violate timing constraints until the end-of-lifetime. Store the circuit (Column #5 in Table 3.11) and its V_{final}.

5. Sign off the same benchmark circuits using different derated libraries characterized with the four combinations: (1) (V_{init}, V_{init}), (2) (V_{init}, V_{max}), (3) (V_{max}, V_{max}), and (4) (V_{init}, V_{final}) obtained from Step (4). This step generates Columns #1 to #4 in Table 3.11.

6. Repeat Step (5) using a derated library with V_{lib} = V_{BTI} = V_{heur1} and = V_{heur2}, where V_{heur1} and V_{heur2} are the predicted V_{final} values obtained with our proposed V_{final} estimation method. We obtain V_{heur1} and V_{heur2} using α = 0 and α = 0.03, respectively, in order to evaluate the results with different α. This step generates Columns #6 and #7 in Table 3.11.

7. Calculate dynamic power of all circuits with AVS (i.e., the AVS mechanism in Figure 3.14) using vectorless analysis in *Synopsys PrimeTime* [254] (input toggle rate is 10%).

Experimental Results

To study potential implications of signoff choices on circuit area and power, we implement circuits with different derated libraries, as well as a reference circuit signed off with V_{lib} = V_{init} and no BTI degradation. The V_{lib} and V_{BTI} of the derated libraries are given in Table 3.11. In Column #1, both V_{lib} and V_{BTI} are set to V_{init}. This setup represents the scenario where the impact of AVS is not considered during library characterization. In Column #2, we set V_{lib} = V_{init} but let V_{BTI} = V_{max} to model the worst-case scenario for use of a derated library.[32] In Column #3, both V_{lib} and V_{BTI} are set to V_{max}. This represents another extreme scenario for the derated library, where the supply voltage of a circuit is assumed to increase to V_{max} to

[32] V_{BTI} = V_{max} means that we calculate $\Delta|V_{th}|$ using Equation (3.14) with V_{gs} = V_{ds} = V_{max}, with the V_{BTI} remaining constant throughout the design lifetime.

compensate for BTI degradation. The setup in Column #4 is similar to that in Column #2 but the V_{BTI} is defined by the V_{final} of the reference circuit. We note that this is an artificial setup because of the dependency between the V_{BTI} and the reference circuit. However, we use this setup to study the impact of ignoring the fact that V_{dd} varies due to AVS, even given that we have a reasonable estimation for BTI degradation. Column #5 in Table 3.11 represents the reference setup, which does not have a specific V_{lib} and V_{BTI} because both voltage values vary over time. Columns #6 and #7 are for the heuristic methods with α = 0 and 0.03, respectively. The values of V_{lib} and V_{BTI} are given in Table 3.9.

Table 3.11: Implementation results with different derated libraries. Circuit lifetime = 10 years. Circuit area and power values are normalized to those of the reference circuits in Column #5.

Circuit #:				1	2	3	4	5	6	7
V_{lib}				V_{init}	V_{init}	V_{max}	V_{init}	N/A	V_{heur1} (α = 0)	V_{heur2} (α = 0.03)
V_{BTI}				V_{init}	V_{max}	V_{max}	V_{final} of #5	N/A	V_{heur1}	V_{heur2}
V_{dd} (V) at 10-year lifetime point	NVT	DC Aging	*C5315*	0.96	0.90	1.10	0.91	1.00	1.01	0.98
			C7552	0.95	0.90	1.10	0.91	1.01	1.03	1.00
			AES	0.92	0.90	1.10	0.90	0.97	0.99	0.96
			MPEG2	0.92	0.90	1.09	0.90	0.97	0.99	0.96
		AC Aging	*C5315*	0.96	0.90	1.10	0.93	0.97	0.99	0.97
			C7552	0.95	0.90	1.10	0.92	0.98	0.98	0.97
			AES	0.92	0.90	1.07	0.91	0.95	0.96	0.94
			MPEG2	0.92	0.90	1.10	0.90	0.95	0.96	0.94
	LVT	DC Aging	*C5315*	0.90	0.90	0.99	0.90	0.91	0.92	0.91
			C7552	0.90	0.90	0.97	0.90	0.90	0.90	0.90
			AES	0.92	0.90	1.09	0.90	0.98	0.99	0.96
			MPEG2	0.93	0.90	1.10	0.90	0.98	0.99	0.97
		AC Aging	*C5315*	0.90	0.90	1.01	0.90	0.90	0.90	0.90
			C7552	0.90	0.90	1.00	0.90	0.90	0.90	0.90
			AES	0.92	0.90	1.10	0.91	0.95	0.96	0.94
			MPEG2	0.93	0.90	1.10	0.92	0.96	0.97	0.95

Figure 3.18 plots the power and area tradeoff for all circuits, where we assume that each circuit increases supply voltage adaptively to compensate for DC BTI degradation. The results show that circuits implemented with different derated libraries have significant differences in power and area. For instance, circuits signed off with the setup in Column #2 of Table 3.11 have up to 38% larger area compared to other circuits. This is because the derated library is

characterized with a worst-case BTI degradation, which leads to pessimistic circuit timing estimation. The results in Table 3.11 show that the V_{dd} of the circuits in Column #2 remain at V_{init} ($0.9V$) at the end of circuit lifetime. This means that AVS is not triggered to compensate for BTI degradation due to the large timing margin that results from a pessimistic signoff criterion. The results also show that some benchmark circuits (*C5315, C7552, AES*) implemented with the setup in Column #2 consume up to 22% more power compared to the reference circuits. This is because the total numbers of instances for the circuits in Column #2 are much larger than for the reference circuits.[33]

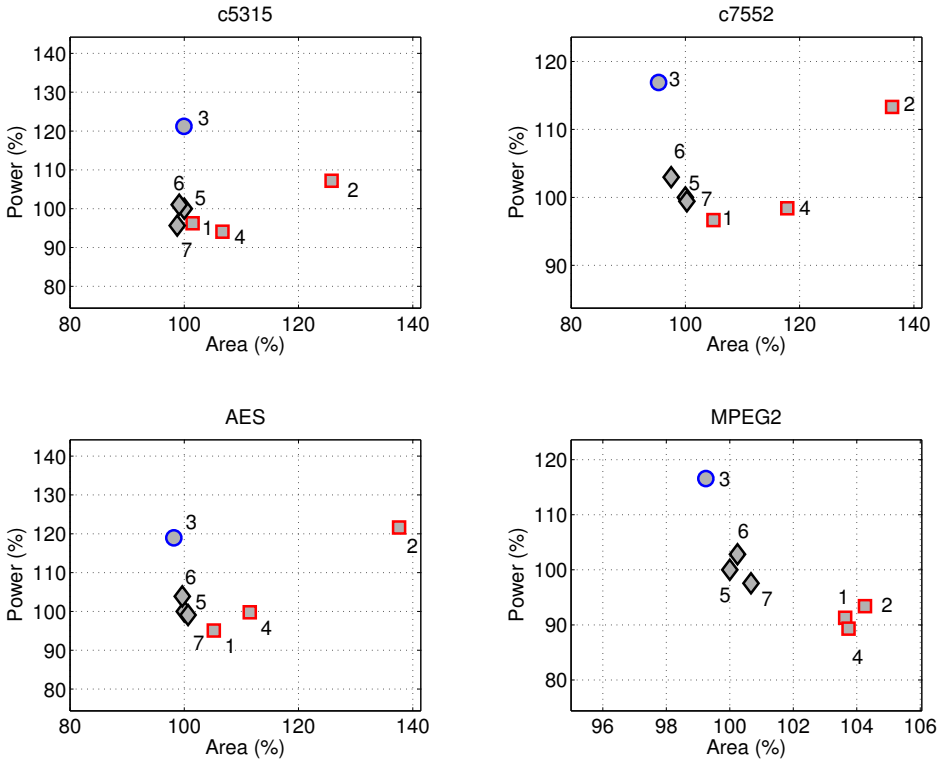
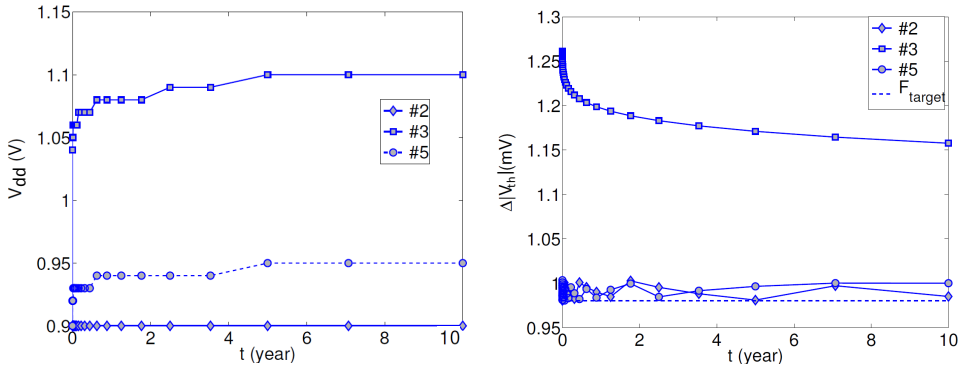


Figure 3.18: Power versus area tradeoff among all circuit implementations (with NVT cells) of each of the four designs, under DC degradation. In each plot, we show the average dynamic power and area of the implementations #1 to #7 for a given design.

Figure 3.18 shows that when more accurate BTI degradation information is available (i.e., implementation #4), the derated library is less pessimistic, which leads to smaller area overheads. However, the circuit areas are 4% to 18% larger than areas of the reference circuits,

[33]For Column #2, the {min, max} overall number of cell instances in the de-noising perturbations are {2397, 2448}, {2741, 2962}, {22883, 23199}, and {25798, 25992} for *C5315, C7552, AES*, and *MPEG2*, respectively. For Column #5, the {min, max} overall number of cell instances in the de-noising perturbations are {2121, 2212}, {2199, 2345}, {17732, 17747}, and {23484, 23985} for the same circuits.



(a) V_{dd} (b) f_{max}

Figure 3.19: V_{dd} and f_{max} of three *MPEG2* circuit implementations obtained with different derated libraries. The V_{dd} of circuit #2 stays fixed at V_{init} because it has large margin for degradation. By contrast, V_{dd} of circuit #3 rises higher than that of circuit #5 soon after manufacturing.

because the derated library does not consider that supply voltage will be higher than V_{init} due to AVS. Since the derated library is pessimistic, the V_{dd} of the circuits in Column #4 remain at V_{init} ($0.9V$) at the 10-year lifetime point (see Table 3.11). Therefore, the circuits in Column #4 have up to 11% lower power compared to the reference circuits.

In the case where the BTI degradation is underestimated and potential V_{dd} increment is ignored (i.e., circuit #1), the inaccurate estimations compensate each other. Therefore, the area and power of the circuits implemented with such a derated library will have only small differences ($< 9\%$) from the corresponding values for the reference circuit. This being said, the quality of results (QoR) of circuits implemented with this derating setup is unpredictable as the outcomes depend on the magnitude of BTI degradation and the sensitivity of circuit performance to AVS.

On the other hand, Figure 3.18 shows that circuits in Column #3 have up to 21% more power compared to the reference circuit. Table 3.11 shows that the V_{dd} of the circuits #3 at 10-year lifetime point is much larger than that of the reference circuit. This indicates that the derated library is optimistic. Therefore, circuits signed off using this derated library will require higher supply voltages to compensate for performance degradation. This shows that an optimistic derated library can cause significant power overhead.

Figure 3.19 shows the V_{dd} and the corresponding f_{max} of the *MPEG2* benchmark circuit over 10 years. When the signoff corner is too optimistic (#3), the implemented circuit fails to meet timing constraints due to BTI degradation. Therefore, the V_{dd} of the circuit is increased to

a higher level than for the reference circuit (#5). On the other hand, the circuits in Column #2 have too much timing margin (no V_{dd} increment over lifetime even if aging) because the signoff corner is too pessimistic.

In Figure 3.18, we can further see that circuits #6 and #7, which are implemented using derated libraries obtained from our heuristic approach, have less than 2% area and less than 4% power difference compared to the reference circuit. This shows that the derated library characterized based on our method can simultaneously capture the effects of the BTI degradation and the varying of V_{dd} due to AVS. Moreover, the circuits can be obtained through a single signoff step, unlike the reference circuits, which require multiple timing analysis and signoff iterations. We also note that the results of #6 and #7 are similar even though the derated libraries have 3% target slack difference. This suggests that our method is not sensitive to small changes in target slack.

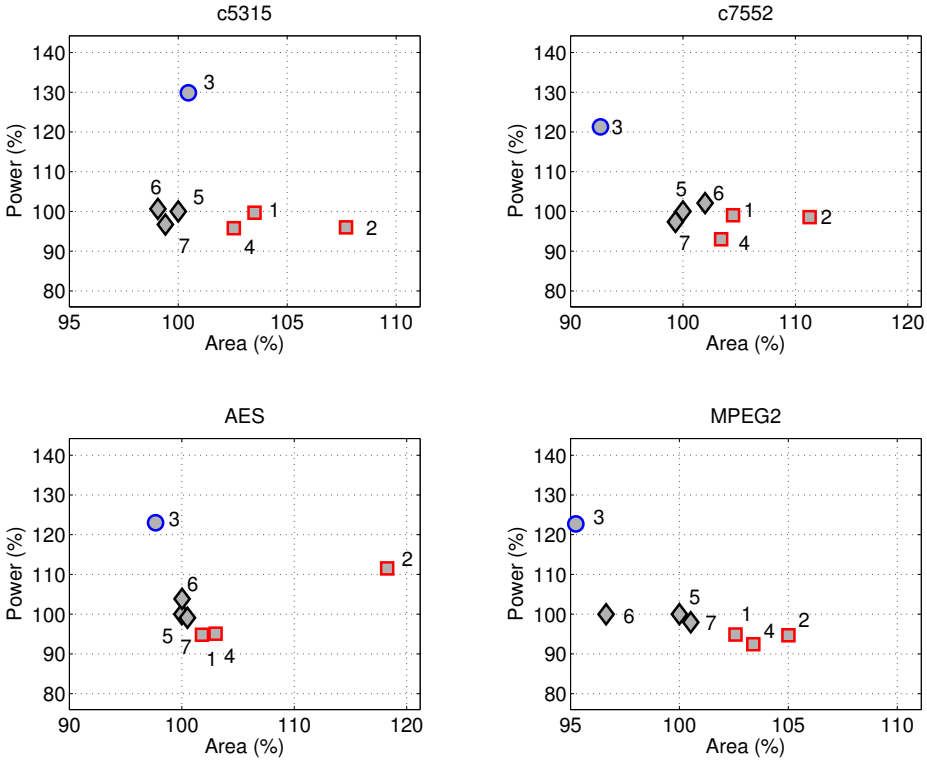


Figure 3.20: Power versus area tradeoff among all circuit implementations (with NVT cells) of each of the four designs, under AC degradation.

Figure 3.20 shows the results of the same experiment setup, but with AC BTI degradation. We see that the results are qualitatively similar to those obtained with DC degradation. Since the AC BTI degradation is about 60% of that in the DC condition, the power/area differences between the circuits are reduced.

Area differences among different *MPEG2* circuit implementations are relatively smaller than those observed for the other three designs, in both AC and DC cases. This is because the ratio of sequential cells (registers) to total cells in the *MPEG2* testcase ($\sim 50\%$) is larger than in the other testcases (e.g., $\sim 20\%$ for *AES* circuit implementations). The main reason for this discrepancy is that we only consider a single size of flip-flop in our characterized library; this enables us to focus on the effect due to combinational cells, which are the main delay contributors of critical paths.

The results in Figures 3.18 and 3.20 show that characterizing a derated library with our proposed method can accurately estimate the effect of BTI aging of a circuit with AVS. The improved estimation can reduce design effort. For example, circuits implemented using the derated libraries #1, #2, #3 and #4 will incur area or power penalty due to inaccurate estimation in BTI aging. Moreover, designers can only discover the inaccuracy after circuit implementation and AVS emulation. Hence, the circuits implemented using an inaccurate derated library may require additional design closure effort (e.g., cycles of sizing, AVS emulation and signoff) and turnaround time to reduce power and circuit area.

3.2.4 Estimation of $V_{critical}$ and Design Margin

As shown in Figure 3.12, an AVS system can increase V_{dd} by at most $V_{max} - V_{init}$ due to the maximum voltage limit. When V_{init} exceeds $V_{critical}$, additional signoff margin is required as the maximum supply voltage increment itself is not sufficient to compensate for BTI-induced circuit delay degradation. To estimate the $V_{critical}$, we apply the heuristics proposed in Section 3.2.2 to approximate the V_{final}. By sweeping the V_{init} from $0.9V$ to $1.1V$ (with step size = $10mV$), we obtain the V_{final} for all timing arcs of 44 cells in the foundry $28nm$ FDSOI standard cell library (NVT and LVT cells). The input slews of the timing arcs are $65ps$, and each cell drives a FO4 load. The target delay is assumed to be 1% lower than the fresh delay at the V_{init}. The lifetime in the simulation is assumed to be 10 years, and we demonstrate both DC and AC results in Figures 3.21(a) and (b), respectively. When the AC BTI stress is applied to the circuits, $V_{critical}$ increases compared to the case of DC BTI stress, indicating that we can use a larger V_{init} without any additional margin due to less aging.

The results in Figure 3.21(a) show that V_{final} (of a cell) reaches V_{max} when V_{init} is higher than $0.96V$. This suggests that we should have an additional signoff margin when the V_{init} is larger than $0.96V$. The margin can be calculated by applying Equation (3.13). Figure 3.22 shows that the worst-case margin (top boundary of the scatter plot) increases rapidly when V_{init}

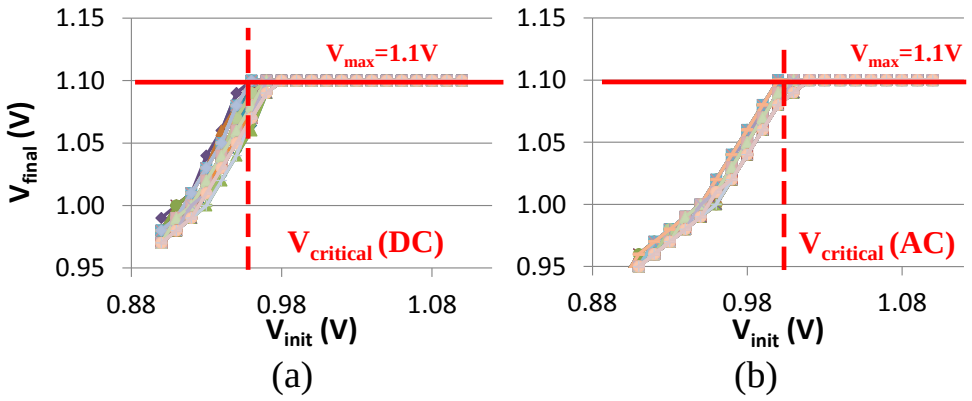


Figure 3.21: The evaluation of $V_{critical}$ for a $28nm$ FDSOI standard cell library. 44 cell types (including LVT and NVT cells) are each connected as cell chains to obtain respective V_{final} versus V_{init} behaviors. (a) DC stress, (b) AC stress.

exceeds $V_{critical}$ ($0.96V$). Therefore, it is necessary for designers to estimate $V_{critical}$. Note that for some cells, the margins on the left-hand side of Figures 3.22(a) and (b) are negative because we apply 1% margin in our AVS emulation. Similar to the observation in Figure 3.21, we see that the required margin is relaxed with AC BTI stress in Figure 3.22(b).

Note that if we do not predict the $V_{critical}$, we need to be more conservative and use a lower V_{init} to ensure that the implemented design can meet the timing constraints. Such conservatism will incur area penalty as design implementations need to meet the same timing constraints at a lower V_{dd}. To quantify the area overhead, we implement designs without any margin (i.e., use non-derated library and zero timing margin) with V_{init} smaller than $V_{critical}$. Figure 3.23 shows that there can be up to 29% area overhead if the V_{init} is $0.080V$ lower than the $V_{critical}$. The area overhead decreases when we use a higher V_{init} and the overhead decreases when we use $V_{init} = V_{critical}$. Although using $V_{init} = 1.020V$ leads to design implementations with smaller area, the designs will fail under DC or AC BTI stress. This means that it is risky to use a high V_{init} without analyzing the $V_{critical}$.

3.2.5 Guardbanding with Derated Libraries and Flat Margins

In Section 3.2.3 above, we have demonstrated the usage of derated libraries. Instead of using derated libraries to guardband design during implementation and final signoff, designers can apply a *flat margin* to all the timing paths in the circuit. The flat margin method is more conservative than the derated library method because the margin is common to all timing paths and

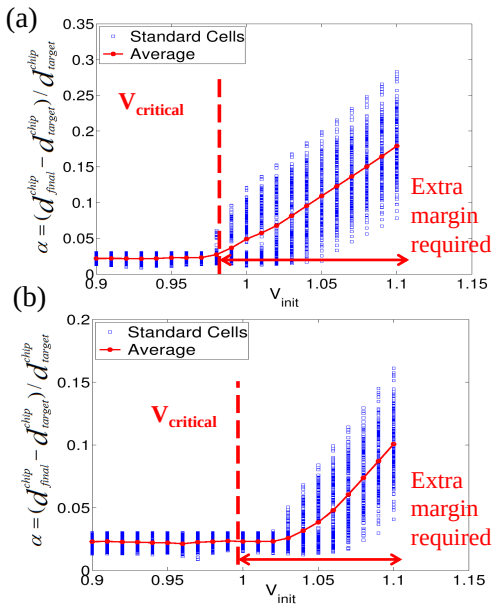


Figure 3.22: Margins (α) required for AVS systems with different V_{init}. Extra margins are required when V_{init} is higher than $V_{critical}$. (a) DC stress, (b) AC stress.

cell types in the circuits. However, the flat margin method can be implemented with minimum changes to the existing signoff flow by tuning the design constraints.[34] In this subsection, we demonstrate how to implement the flat margin method with our heuristics in Section 3.2.3, then compare circuit implementations signed off with a flat margin against implementations signed off with derated libraries.

Implementation of Flat Margin Method and Comparison with Derated Library Method

To obtain the aged delays of circuits, we obtain cell libraries with the device model from the foundry $28nm$ FDSOI PDK. The libraries are characterized with different sets of $\{V_{dd}$, ΔV_{thp}, $\Delta V_{thn}\}$ using Synopsys SiliconSmart [256]. 48 libraries in this technology node are characterized for the delay calculation. The delay calculation steps are similar to those described in Section 3.2.1. We implement three OpenCores circuits [243] (*AES*, *MPEG2*, and *JPEG*) with Synopsys Design Compiler [250] and IC Compiler [252]. The nominal clock periods of *AES*, *MPEG2*, and *JPEG* are $600ps$, $650ps$, and $960ps$, respectively. We consider both DC and AC aging and circuit lifetime $=$ 10 years. The implementations for both methods (the flat margin

[34] To our understanding, the use of derated ("10-year") libraries, prevalent in the $65nm$ node era, has been largely supplanted by flat margin methodologies in the $28nm$ era. Our study and results raise interesting questions about potential suboptimality of this industry trend.

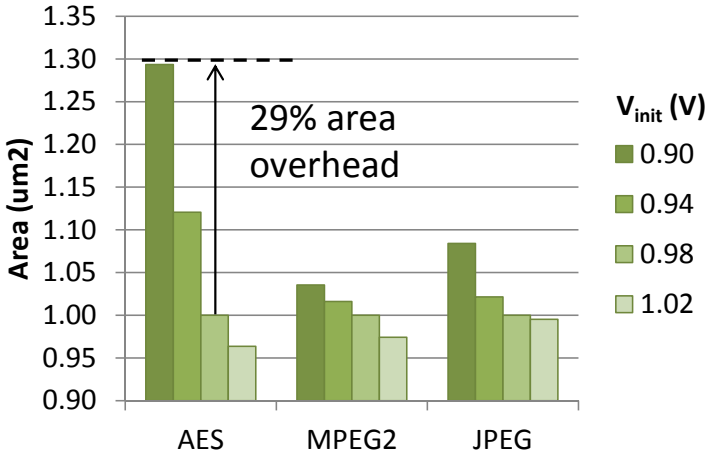


Figure 3.23: Area of circuits implemented with non-derated library and zero timing margin. There are area overheads when V_{init} is lower than $V_{critical}$ = 0.98V.

and derated library methods) are described below. After these implementations, the delay and power of these circuits are calculated in Matlab programs.

To guarantee that the circuits can still properly function at the end of lifetime, we use $V_{lib} = V_{final}$ for signoff. Because V_{final} of circuits is also required to obtain the delay and aging at the end of lifetime, there exists a similar "chicken and egg" loop in the flat margin method. To overcome this, we use the heuristic in Section 3.2.2 to estimate V_{final} (i.e., using the simulated V_{final} from cell chains) and then apply it to Equation (3.15) to calculate the required *clock constraint* for circuit implementation. The STA results show that these implementations of the flat margin method have no timing violation in Table 3.12, which validates our implementation approach. We use

$$\text{clock constraint} = (\text{nominal clock period}) \cdot [1 - \frac{d^{chip}_{final} - d^{chip}_{fresh}(V_{final})}{d^{chip}_{final}}] \tag{3.15}$$

where $d^{chip}_{fresh}(V_{final})$ is the delay of a circuit without aging when V_{dd} = V_{final}. d^{chip}_{final} is the delay of a cell with aging at the end of lifetime.

We use the heuristics from Section 3.2.3 to sign off circuits using derated libraries. The derated libraries are characterized with V_{lib} = V_{BTI} = V_{final}, with the V_{final} obtained from the cell chain simulation. Because the derated libraries have already considered aging, the timing constraints are set to nominal clock periods without additional margins.

Table 3.12: Area and average power results from methods (1) with flat margin, and (2) with derated libraries. The numbers under the design names are nominal clock periods. The nominal clock periods of *AES*, *MPEG2*, and *JPEG* are 600ps, 650ps, and 960ps, respectively.

		DC						AC					
		From cell chain		With flat margin		Ratio of ($\frac{\text{Derated}}{\text{Flat}}$)		From cell chain		With flat margin		Ratio of ($\frac{\text{Derated}}{\text{Flat}}$)	
	V_{init} (V)	Margin (ps)	V_{final} (V)	Power (mW)	Area (μm^2)	Power	Area	Margin (ps)	V_{final} (V)	Power (mW)	Area (μm^2)	Power	Area
AES	0.90	78	0.99	40.0	18349	0.9400	0.9576	42	0.94	39.3	19205	0.9771	0.9683
	0.94	108	1.06	39.7	18215	0.9498	0.8738	54	0.99	37.0	17169	0.9572	0.9496
	0.98	162	1.10	40.6	18248	0.9273	0.8500	72	1.06	35.3	15164	0.9972	0.9557
	1.02	168	1.10	41.6	18166	0.9091	0.8538	96	1.10	35.6	15135	1.0122	0.9223
	1.06	168	1.10	44.2	18166	0.8833	0.8538	96	1.10	37.9	15135	0.9526	0.9223
	1.10	168	1.10	47.2	18166	0.8716	0.8538	96	1.10	40.7	15135	0.9168	0.9223
MPEG2	0.90	85	0.99	34.5	24178	0.9717	0.9850	46	0.94	30.9	24094	0.9917	0.9714
	0.94	117	1.06	36.3	24414	1.0051	0.9741	59	0.99	32.7	23083	1.0155	1.0124
	0.98	176	1.10	32.0	23410	1.1555	0.9966	78	1.06	34.7	22986	1.0625	0.9950
	1.02	182	1.10	34.8	23880	1.0675	0.9770	104	1.10	34.7	22616	1.0782	0.9948
	1.06	182	1.10	37.6	23880	1.0218	0.9770	104	1.10	35.8	22616	1.0475	0.9948
	1.10	182	1.10	40.2	23880	1.0107	0.9723	104	1.10	38.4	22616	0.9979	0.9984
JPEG	0.90	125	0.99	53.4	65387	0.9875	0.9594	67	0.94	50.2	64461	1.0201	0.9917
	0.94	173	1.06	55.3	64788	1.0546	0.9433	86	0.99	54.1	63528	0.9777	0.9745
	0.98	259	1.10	53.7	66158	1.1054	0.9343	115	1.06	55.8	61471	1.0181	0.9829
	1.02	269	1.10	58.0	66928	1.0238	0.9236	154	1.10	56.1	61043	1.1315	1.0122
	1.06	269	1.10	62.4	66928	0.9806	0.9236	154	1.10	59.3	61043	1.0729	1.0122
	1.10	269	1.10	66.5	66928	0.9689	0.9236	154	1.10	63.5	61043	1.0234	1.0122

Experimental Results

From the results in Table 3.12, we have the following observations: (i) Circuits signed off using the flat margin method have up to 15% larger area compared to those signed off using derated libraries. This is because the flat margin method determines the signoff margin based on the worst timing arc in the cell library, while the derated library has differently aging cells and arcs. (ii) When V_{init} = V_{max}, the derated library method shows a power benefit in testcases *AES* and *JPEG*, with both DC and AC degradation; this is because the larger areas due to the pessimism in (i) also result in higher power. There is no power benefit for the *MPEG2* testcase because the total power is dominated by the internal power of sequential cells (registers), which varies with the transition time of timing arc. (iii) When AVS has more headroom to adjust the V_{dd} (i.e., $V_{max} - V_{init}$ is larger), we can observe that power disadvantage of the flat margin method lessons. This is because the derated library method is less pessimistic, and the V_{dd} will increase faster than with the flat margin method when $V_{max} - V_{init}$ is larger.

These observations lead to the following summary. (i) Both derated library and flat margin methods are pessimistic about the aging, which indicates that both methods are usable for signoff. (ii) The flat margin method has the advantage of simplicity because it can be implemented by tuning the timing constraints in the existing signoff flow. We propose that our V_{final} estimation heuristic be used to obtain the flat margin in Section 3.2.5. (iii) However, the flat margin is more pessimistic than the derated library method, so it results in larger area penalties.

3.2.6 Conclusions

We analyze aging-aware timing signoff issues for circuits with AVS. Based on our analysis in Section 3.2.4, V_{init} must be smaller than $V_{critical}$ or additional margin is required. As discussed in Section 3.2.4, $V_{critical}$ can be estimated through our proposed heuristics. And, when margin is required there are two signoff methods: (i) using derated libraries or (ii) applying flat margins.

When guardbanding aging with derated libraries, there are discrepancies among the voltages that are applied for derated library characterization, and the voltage through lifetime of a circuit with AVS – namely, V_{lib}, V_{BTI} and V_{final}. Inconsistency among these voltages can cause the derated library to be either optimistic or pessimistic with respect to the impact of BTI degradation and AVS. To avoid the design overhead that potentially arises from poor selection of V_{lib} and V_{BTI} during library characterization, we propose a library characterization heuristic which suggests that $V_{lib} = V_{BTI} \approx V_{final}$ is the best strategy for derated library characterization. We also propose a method to estimate the V_{final} from replica circuits and AVS parameters, which are both available early in the design process.

With the V_{final} heuristic, we provide an implementation example for the flat margin method in Section 3.2.5. Although the flat margin and derated library methods can both guarantee timing correctness under aging, we demonstrate in a foundry $28nm$ FDSOI technology that there can be up to 15% area overhead associated with the flat margin method compared to the derated library method.

3.3 BEOL Corner Optimization

In a conventional implementation methodology, designers sign off an SoC design at extreme PVT conditions to ensure functional correctness. As wire geometries continue to shrink with each new process node, wire resistance (R) and capacitance (C) have become major sources of variation [155], which must be accounted for by signoff at BEOL corners. In current industry-

standard signoff methods, *conventional BEOL corners* (CBCs) are defined such that all BEOL layers vary in the same way [91]. For example, Table 3.13 (see Section 3.3.1) shows common BEOL corners in which the wire width (ΔW), wire thickness (ΔT) and dielectric thickness (ΔH) variations are biased to the minimum or maximum values.[35] Although BEOL parameters have strong spatial correlations within a die [139], different BEOL parameters are not fully correlated [77] [91] [136] [146] [211]. When the parameters are not fully correlated, the likelihood of a worst-case (or best-case) condition on all layers is vanishingly small (if not a physical impossibility). Therefore the CBCs are unnecessarily pessimistic, which results in longer chip implementation schedules (time spent on design closure steps).

To reduce the pessimism in CBCs, various statistical RC extraction and timing analysis methods have been proposed [1] [57] [72]. The main drawback of statistics-based method is the lack of availability of commercial EDA tools to characterize a RC variation model (e.g., sensitivities of RC to BEOL physical parameters). Although we can construct the RC variation model by extracting RC at nominal and perturbed corners for each variation source [72], this method requires a lot of computing resources. For example, to characterize an interconnect stack with nine metal layers and three variation sources per layer, we need 28 RC extractions for a nominal corner and 27 perturbed corners. Moreover, the extracted parasitics are design-specific and they must be updated when the design changes.

Alternatively, Lu and McCullen [138] propose a BEOL variation-aware timing analysis method based on a layout-to-SPICE netlist extraction tool. Since the extraction tool can annotate the nominal RC value as well as the bounds of RC in the SPICE netlist, the BEOL-induced timing variation can be simulated using SPICE. However, the SPICE-based timing analysis is slower than static timing analysis (STA), and commercial extraction tools do not have the option to extract and annotate BEOL parameters into a netlist.

For corner-based timing analysis, there are methods to find the worst-case BEOL variation scenarios [91] [153] [188], but these scenarios are far from the typical BEOL variations seen in IC manufacturing. Thus, signing off a design using these BEOL variation scenarios will incur large design overheads [95]. Yamada and Oda [211] propose a simple method to tighten BEOL corners based on the wirelengths of BEOL layers. This corner-based method has the advantage that statistical extraction is only required once per technology for validation. However, this approach is oversimplified in that the estimation may be optimistic when path delays have different and opposite sensitivities to BEOL variations.

[35] The ΔW, ΔT and ΔH in Table 3.13 are extracted from foundry's BEOL corners. The definitions of the BEOL corners match with those described in [123].

In this section, we propose a signoff methodology with *tightened BEOL corners* (TBCs) to reduce the impact of pessimism in CBCs. Our method is based on an observation similar to [211], i.e., the wires on timing-critical paths are typically routed through different BEOL layers. For example, Figure 3.24 shows that the wirelength ratio of (setup) critical paths extracted from a design are mostly routed on layers M2 to M6. Figure 3.25 shows that, for 92% of the paths, the maximum wirelength from a single layer is less than 60% of the total wirelength. When process variations of the BEOL layers are not fully correlated, the timing variation on a critical path is typically much smaller than that estimated using CBCs due to averaging of uncorrelated variations.[36] Our analysis (see Section 3.3.2) shows that the delay variation at a CBC (with respect to the typical BEOL condition) can be much larger than the delay variation obtained from a statistical analysis. Further, we observe that the pessimism of a CBC depends on the sensitivities of critical-path delays to resistance and capacitance variations. Our results also show that CBCs have small or no pessimism for certain kinds of critical paths. Thus, we cannot apply TBCs to the entire design as suggested in [211]. To address this issue, we propose to choose the signoff corners (i.e., CBCs or TBCs) for each path based on its delay sensitivities to resistance and capacitance. By using this method, we can safely sign off a path using TBCs or CBCs without underestimating the delay variation of the paths.

Our main contributions are as follows.

- We show that the pessimism of a CBC depends on the sensitivities of critical-path delay to BEOL resistance and capacitance, and that the trend is similar across different designs.
- We propose a method to identify the critical paths which can use tightened BEOL corners for signoff. We show that this method can reduce the number of paths with timing violations by up to 100% and improve WNS and TNS by up to $101ps$ and $53ns$, respectively.

3.3.1 BEOL Variation Model

We denote the index of a metal layer in an interconnect stack by m and the total number of metal layers by N_{layer}. We denote the conductor width and thickness of the layer m by W_m and T_m, respectively. Similarly, we denote the spacing between conductors for layer m by S_m, and the thickness of the layer's inter-layer dielectric (i.e., the distance between layer m and layer $m + 1$) by H_m. Figure 3.26 illustrates an example of the interconnect stack with three metal layers (M1, M2 and M3).

[36] As explained in [91], given a timing path, it is possible to find a worst-case BEOL scenario for which the delay estimated at the worst-case BEOL scenario is worse compared to those at CBCs. However, the worst-case BEOL scenario is rare or else not significant enough to cause timing violations in actual chips.

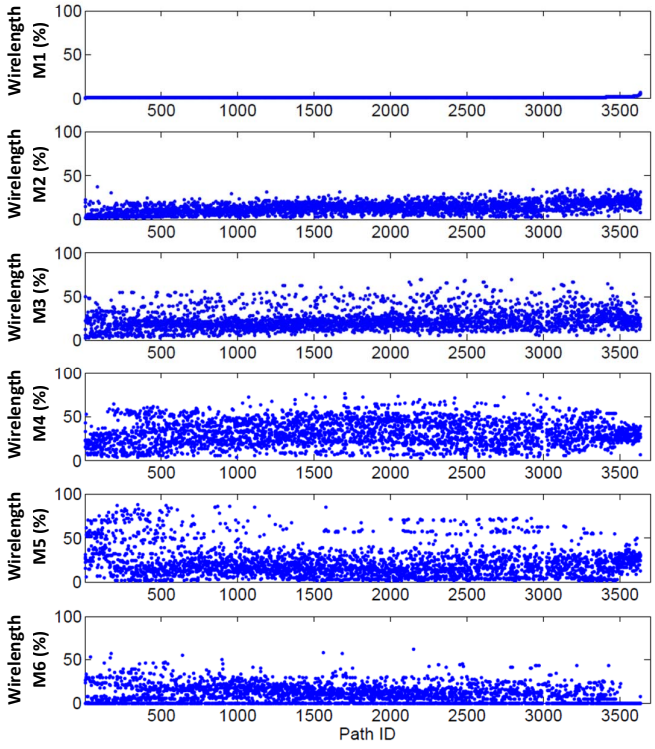


Figure 3.24: Wirelength distribution of critical paths on different BEOL layers.

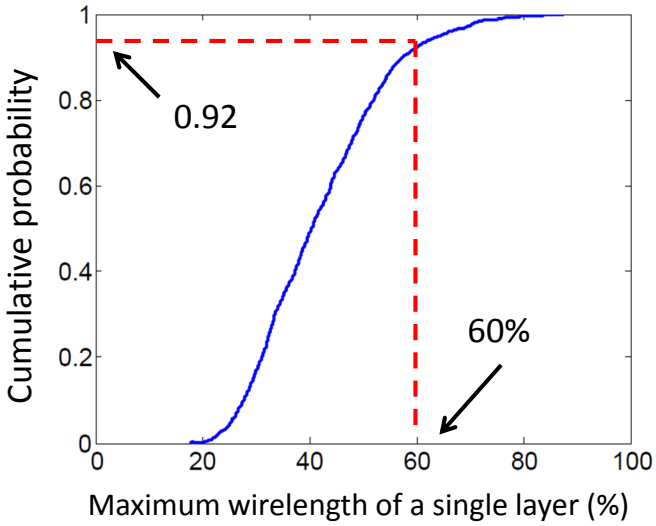


Figure 3.25: Cumulative probability of the maximum wirelength percentage of a single layer (relative to total wirelength on its corresponding path).

Conventional BEOL Corners

The major variation sources in a BEOL corner are ΔW_m, ΔT_m and ΔH_m, which correspond to the variations in W_m, T_m, and H_m, respectively.[37] A CBC is modeled by biasing the variation sources in a BEOL technology file (e.g., itf [253] or ict [227]). For example, Table 3.13 shows the ΔW_m, ΔT_m and ΔH_m for typical CBCs. Note that the ΔW_m, ΔT_m and ΔH_m are biased in the same way for all layers in a CBC. It should also be noted that the RC-best (Y_{rcb}) and C-worst (Y_{cw}) corners have similar ΔW and ΔT. Meanwhile, the RC-worst (Y_{rcw}) and C-best (Y_{cb}) corners have similar ΔW and ΔT. Thus, the wire resistance extracted at Y_{rcb} and Y_{cw} (resp. Y_{rcw} and Y_{cb}) are similar but the capacitance is larger (resp. smaller) at Y_{cw} (resp. Y_{cb}) because of a smaller (resp. larger) inter-layer dielectric thickness.

Table 3.13: Typical BEOL corners with skewed parameters.

Corner	ΔW_m	ΔT_m	ΔH_m
Y_{typ}	typical	typical	typical
Y_{cb}	minimum	minimum	maximum
Y_{cw}	maximum	maximum	minimum
Y_{rcb}	maximum	maximum	maximum
Y_{rcw}	minimum	minimum	minimum

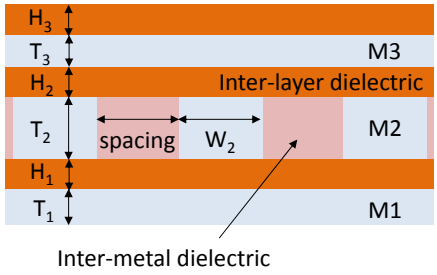


Figure 3.26: Illustration of the cross-section of a typical metal stack.

[37] Spacing variation is implicitly defined by the ΔW_m.

Tightened BEOL Corners

We denote a tightened BEOL corner by Y_{ref_α}, where α is a scaling factor and Y_{ref} is a CBC, i.e., $Y_{ref} \in \{Y_{cb}, Y_{cw}, Y_{rcb}, Y_{rcw}\}$. We define ΔW_m, ΔT_m and ΔH_m of a Y_{ref_α} as

$$\begin{aligned} \Delta W_m \text{ of } Y_{ref\_\alpha} &= \alpha \cdot \Delta W_m \text{ of corner } Y_{ref} \\ \Delta T_m \text{ of } Y_{ref\_\alpha} &= \alpha \cdot \Delta T_m \text{ of corner } Y_{ref} \\ \Delta H_m \text{ of } Y_{ref\_\alpha} &= \alpha \cdot \Delta H_m \text{ of corner } Y_{ref} \end{aligned} \tag{3.16}$$

Statistical BEOL Variation

For an interconnect stack with N_{layer} layers, there are $3N_{layer}$ variation sources. We model each of these variation sources as a Gaussian random variable z_v ($v = 1, 2, ..., 3N_{layer}$). The correlations among the random variables are defined by a correlation matrix (Σ). Since BEOL parameters are correlated if they are fabricated using the same *process module* [146], we model the correlation between two variance sources as follows.

$$\Sigma_{u,v} = \begin{cases} 1 \text{ if } u = v \\ \gamma \text{ if both } z_u \text{ and } z_v \text{ are } \Delta W,\ \Delta H \text{ or } \Delta T \\ \quad \text{of different BEOL layers and the layers are in} \\ \quad \text{the same process module.} \\ 0 \text{ otherwise} \end{cases} \tag{3.17}$$

where $\Sigma_{u,v}$ is the entry at the u^{th} row and v^{th} column in Σ. γ is the correlation between z_u and z_v. Due to the lack of actual manufacturing data, we assume that γ is the same for different pairs of variation sources. In our experiments, we study two scenarios with γ = 0.5 [146] and $\gamma = 0.0$ (i.e., all variation sources are independent). Unless otherwise specified, the following statistical analyses use $\gamma = 0.0$. For the nine-layer interconnect stack in our experiment, there are three process modules:

- Layers M1, M2 and M3 $\in$ process module 1
- Layers M4, M5, M6 and M7 $\in$ process module 2
- Layers M8 and M9 $\in$ process module 3

We define Y_v as the BEOL corner in which only the v variation source is perturbed by one standard deviation from the typical condition.[38] We extract the delay sensitivity of the j^{th} path (p_j) to the v^{th} variation source ($\Delta d_{j,v}$) by using the finite-difference method [72].[39]

$$\Delta d_{j,v} = d_j(Y_v) - d_j(Y_{typ}) \tag{3.18}$$

where Y_{typ} is the typical BEOL corner. $d_j(Y_v)$ and $d_j(Y_{typ})$ are, respectively, the delay of p_j at Y_v and Y_{typ}. Note that the layout-induced RC variation is accounted for in the RC extraction. The BEOL-induced delay variation for p_j (σ_{path_j}) is given by the following equation.

$$\begin{aligned} \sigma_{path\_j} &= \sqrt{\sum_{v=1}^{3N_{layer}} (\Delta d'_{j,v})^2} \\ \text{where } [\Delta d'_{j,3N_{layer}}, ..., \Delta d'_{j,3N_{layer}}] &= [\Delta d_{j,1}, ..., \Delta d_{j,3N_{layer}}] \cdot \lambda \\ (\lambda \cdot \lambda^T) &= \Sigma \end{aligned} \tag{3.19}$$

We decompose Σ to obtain λ by using the *Cholesky decomposition* method. λ is a lower triangular matrix and λ^T is the transpose of λ.

Note that the delay variation is also affected by the drive strength of standard cells which has within-die random variation [167]. Therefore, the delay variation of different nets on the same metal layer may not be fully correlated. Since our variation model assumes that the delay variation on a single metal layer is fully correlated, we may underestimate the effect of averaging random variations.

3.3.2 Pessimism in Conventional BEOL Corners

Unlike hold-time violations which can be fixed by buffer insertion, fixing a setup timing-critical path at CBC corners has become a very challenging task due to the increased wire resistance and BEOL variation. For example, increasing the drive strengths of standard cells along a setup timing-critical path is a typical approach to fix a setup-time violation. However, when the path is dominated by wire delay (e.g., a path with relatively long wires), increasing the drive strengths of cells can only reduce a fraction of the path delay, which may be insufficient to fix the setup timing violation. This problem is even more critical at high V_{dd} and/or high temperature operating conditions in which the impact of wire delay variation is more significant. In the

[38] We assume that the ΔW_m, ΔH_m and ΔT_m in the Y_{rcb} and Y_{rcw} corners correspond to $+3$ and -3 standard deviations, respectively.

[39] We assume that the path delay varies linearly with variation sources [1].

following discussion, we only focus on reducing the pessimism of CBC on the data path of setup timing-critical paths.[40]

We define $\Delta d_j(Y)$ as the difference between the delays of p_j at corners Y and Y_{typ}, i.e., $\Delta d_j(Y) = d_j(Y) - d_j(Y_{typ})$. We consider p_j as "safe" if the path is signed off at a corner Y, for which $\Delta d_j(Y)$ is larger than $3\sigma_{path\_j}$.

$$\exists\, Y, \Delta d_j(Y) \geq 3\sigma_{path\_j} \tag{3.20}$$

Our goal is to find the tightened BEOL corners such that the design signed off using these corners will meet the safe condition in Equation (3.20). Meanwhile, the corners should not be overly pessimistic, i.e., the difference between $\Delta d_j(Y)$ and $3\sigma_{path\_j}$ should be minimized.

Analysis

When BEOL variations are small, path delay variations can be approximated as a linear function of BEOL variations [1]. Based on this assumption and the definition of the TBC in Equation 3.16,

$$\Delta d_j(Y_\alpha) = \alpha \cdot \Delta d_j(Y) \tag{3.21}$$

where $\Delta d_j(Y_\alpha)$ is the delay variation at a TBC. To satisfy the safe condition at a Y_α, the smallest scaling factor for p_j ($\alpha_j(Y)$) is given by

$$\alpha_j(Y) = \frac{3\sigma_{path\_j}}{\Delta d_j(Y)} \tag{3.22}$$

Figure 3.27 shows the scaling factors of a set of critical paths for Y_{cw} and Y_{rcw}. The figure shows that $\alpha_j(Y)$ is small when $\Delta d_j(Y)$ is large but increases rapidly when $\Delta d_j(Y)$ approaches zero. Also, there are paths for which their $\Delta d_j(Y_{cw})$ (resp. $\Delta d_j(Y_{rcw})$) become negative. This happens because Y_{cw} (resp. Y_{rcw}) corner has smaller parasitic resistance (resp. capacitance) and the paths are more sensitive to the changes in resistance (resp. capacitance). The results also imply that we need to sign off at both Y_{cw} and Y_{rcw} corners to capture the impact of interconnect variation. When we analyze both Y_{rcw} and Y_{cw} corners, the paths which have a smaller $\Delta d_j(Y_{cw})$ will have a larger $\Delta d_j(Y_{rcw})$, and vice-versa for the paths which have larger $\Delta d_j(Y_{cw})$. Thus we should only consider the α_j at the *dominant corner* which has a larger

[40] Our signoff methodology is not applicable to the hold critical paths because there is not much averaging effect in the short data paths. Also, pessimisms of the CBCs is not significant for the clock network which is typically implemented on a few BEOL layers.

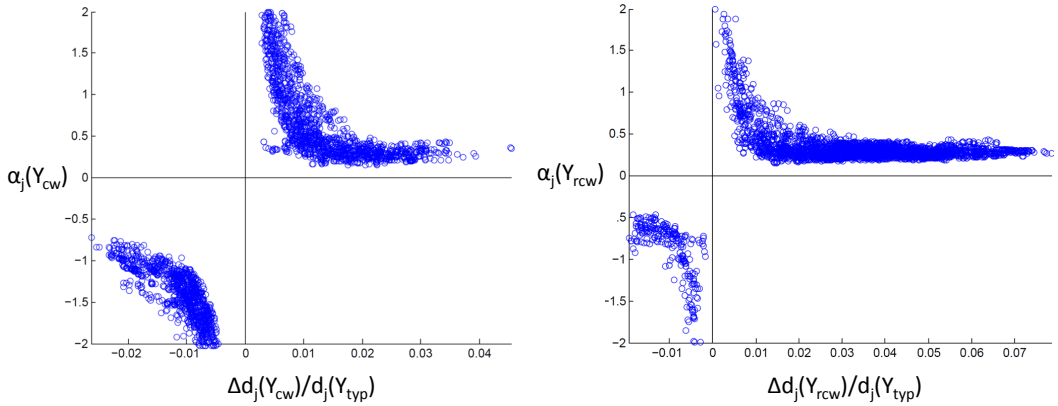


Figure 3.27: α_j versus Δd_j for critical paths obtained from the *NETCARD* benchmark circuit.

$\Delta d_j(Y)$. The actual scaling factor (α_j^{act}) is defined as

$$\alpha_j^{act} = \frac{3\sigma_{path\_j}}{max(\Delta_{delay\_j,Y_{cw}}, \Delta_{delay\_j,Y_{rcw}})} \tag{3.23}$$

To understand the trends in Figure 3.27, we analyze the relationships between σ_{path_j} and $\Delta d_j(Y)$. Figure 3.28 shows that there is a strong correlation between $3\sigma_{path\_j}$ and $\Delta d_j(Y)$. Moreover, most of the paths have a α_j^{act} smaller than 0.5. The small α_j^{act} is due to the averaging of uncorrelated variations when the wires along the paths are routed on many metal layers.

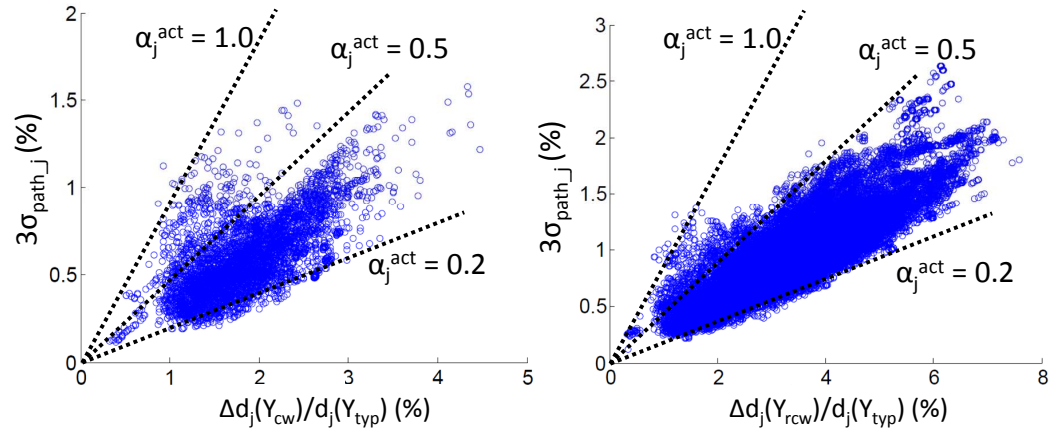


Figure 3.28: $3\sigma_{path\_j}$ versus $\Delta d_j(Y)$.

Figure 3.29 shows the relationships between α_j^{act}, $\Delta d_j(Y_{cw})$ and $\Delta d_j(Y_{rcw})$. Each circle in the figure represents a path, the coordinates of a circle on the x- and y-axes indicate its (normalized) $\Delta d_j(Y_{cw})$ and $\Delta d_j(Y_{rcw})$. Meanwhile, the color of the circles indicates the

magnitude of α_j^{act}. From the figure, we can see that the paths with a large α_j^{act} have small $\Delta d_j(Y_{cw})$ and $\Delta d_j(Y_{rcw})$, e.g., both $\Delta d_j(Y_{cw})$ and $\Delta d_j(Y_{rcw})$ are smaller than 0.03 when α_j^{act} is larger than 0.5.

Our analysis shows that the paths with a large α_j^{act} have similar delay sensitivities to R and C. Since a CBC is biased such that the R and C change in opposite directions (with respect to Y_{typ}, the total delay variation at a CBC is very small for the paths with similar delay sensitivities to R and C. In other words the delay variation due to R and C are cancelled out. Note that the cancellation effect is an artifact of CBCs, which does not exist in the statistical RC analysis. Thus, $3\sigma_{path\_j}$ is larger than the delay variation at a CBC (i.e., α_j^{act} is large) for this kind of path.

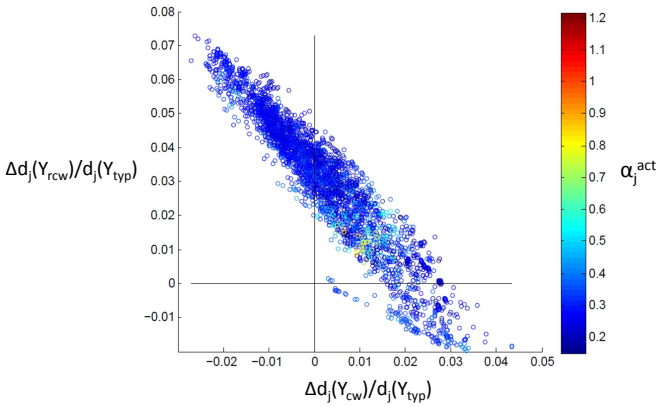


Figure 3.29: α_j^{act} versus Δd_j at Y_{cw} and Y_{rcw} corners.

Since the α_j^{act} is mainly affected by $\Delta d_j(Y_{cw})$ or $\Delta d_j(Y_{rcw})$, we propose to classify the critical paths based on their $\Delta d_j(Y)$.

$$p_j \in \begin{cases} G_{TBC} \text{ if } [(\Delta d_j(Y_{rcw}) > A_{rcw}) \text{ or } (\Delta d_j(Y_{cw}) > A_{cw})] \\ G_{CBC} \text{ otherwise} \end{cases} \tag{3.24}$$

G_{CBC} and G_{TBC} are respectively the set of paths to be signed off using CBC and TBC. A_{rcw} and A_{cw} are, respectively, the thresholds for the $\Delta d_j(Y_{rcw})$ and $\Delta d_j(Y_{cw})$, which determine whether a path is in G_{TBC} or G_{CBC}.

Proposed Method

Figure 3.30 describes our signoff methodology. Given a routed design, we first analyze the data paths at Y_{cw}, Y_{rcw} and Y_{typ} to classify the setup timing-critical paths into G_{TBC} or

G_{CBC}. The paths in G_{TBC} (resp. G_{CBC}) will be analyzed using TBC (resp. CBC). If there are timing violations, the paths are fixed through a path-based ECO at the corresponding BEOL corners. The design is closed when there are no paths with timing violations in both G_{TBC} and G_{CBC}.

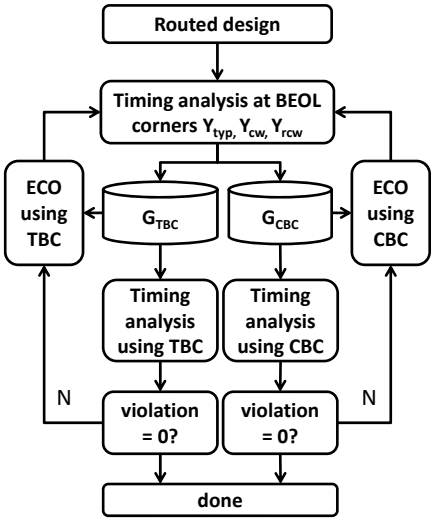


Figure 3.30: Proposed signoff flow.

Based on our experimental results (see Section 3.3.3), we observe that the critical paths of the designs implemented using the same technology and design flow have similar structures. Therefore, we propose to extract the values of A_{cw} and A_{rcw} from a set of representative critical paths and use them for other designs implemented using the same technology and design flows. By using this approach, we only need to perform the costly statistical analysis to characterize A_{cw} and A_{rcw} when there is a major change in the technology or design flow.

Given a set of representative critical paths as well as their corresponding timing constraints and operating conditions, the problem is to select the A_{cw}, A_{rcw} and TBCs to minimize the pessimism in CBCs while satisfying the safe condition in Equation (3.20). To solve this problem, we perform a statistical analysis and extract the *optimal scaling factors* ($\alpha^{opt}(Y_{rcw})$ and $\alpha^{opt}(Y_{cw})$) for different A_{cw} and A_{rcw}.[41]

$$\begin{aligned} \alpha^{opt}(Y_{rcw}) &= \max_j(\alpha_j^{act}(Y_{rcw})), \Delta d_j(Y_{rcw}) > A_{rcw} \\ \alpha^{opt}(Y_{cw}) &= \max_j(\alpha_j^{act}(Y_{cw})), \Delta d_j(Y_{cw}) > A_{cw} \end{aligned} \tag{3.25}$$

[41]The $\alpha^{opt}(Y_{cw})$ (resp. $\alpha^{opt}(Y_{rcw})$) is optimal for a given set of representative critical paths, along with a threshold value A_{cw} (resp. A_{rcw}).

Figure 3.31 shows that as $\alpha^{opt}(Y_{rcw})$ (resp. $\alpha^{opt}(Y_{cw})$) reduces, the A_{rcw} (resp. A_{cw}) increases but the $|G_{TBC}|$ reduces. In other words, as we tighten a BEOL corner, the number of paths which can be signed off using the TBC reduces.

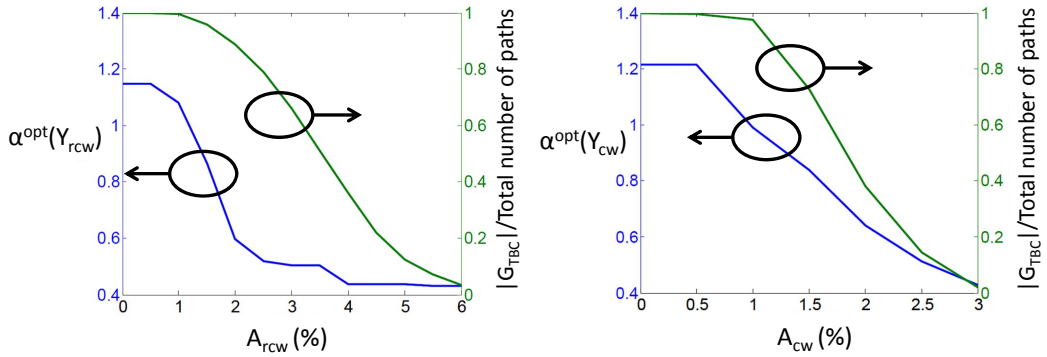


Figure 3.31: Tradeoff between $A_{rcw,cw}$ and $|G_{TBC}|$ with $\gamma = 0.0$.

3.3.3 Experimental Results

We use three designs from ISPD contests [165] [200] and the *OpenCores* [243] as the testcases in our experiments. The designs are placed and routed with a triple-V_{th} $45nm$ foundry library using *Synopsys IC Compiler* [252]. To emulate the highly resistive BEOL in advanced technology, we scale the resistivity in the BEOL model file by $8\times$. For timing signoff, we use *Synopsys PrimeTime* [254]. The PVT condition for setup timing analysis is $SS, 0.90V$ and $125°C$. We use the Y_{cw} and Y_{rcw} during the implementations. The key design parameters of the implemented testcases are listed in Table 3.14.

Table 3.14: Physical implementation results of testcases.

	LEON3MP	*NETCARD*	*SUPERBLUE12*
Clock period (ns)	1.80	2.00	3.10
Gate count	232K	575K	1031K
Utilization (%)	84	79	82
Core area (mm^2)	0.45	1.04	1.91
Max Transition (ns)	0.33	0.33	0.33

Experiment Setup

After placement and routing, we fix the timing violations in the designs by using the *fix_eco* commands in *Synopsys PrimeTime* [254] until there are no improvements. Then we extract 1000 setup timing-critical paths at Y_{cw} and Y_{rcw}, separately. To emulate our signoff methodology, we filter the extracted paths based on the definition in Equation 3.25 to obtain G_{tbc}. For our signoff methodology, the paths in G_{TBC} are analyzed using Y_{cw_α} and Y_{rcw_α}. Meanwhile, the paths in G_{CBC} are analyzed using Y_{cw} and Y_{rcw}. In our experiments, we set $\alpha^{opt}(Y_{rcw})$ equal to $\alpha^{opt}(Y_{cw})$.[42] The A_{rcw} and A_{cw} for different α^{opt} and statistical BEOL models are listed in Table 3.15. To collect the representative timing-critical paths, we implemented another *NETCARD* benchmark circuit with a clock period = $2.3ns$ and extract the top 10000 paths at Y_{rcw} and Y_{cw}. Note that the critical paths are different from that of the *NETCARD* testcase described in Table 3.14. Since the representative timing-critical paths can be different from the actual testcases, we increase the values A_{rcw} and A_{cw} by 1% to account for the sampling error in the construction of the representative paths.

Table 3.15: Configurations for TBC-based signoff.

Configuration	α^{opt}	$\gamma = 0.0$		$\gamma = 0.5$	
		A_{cw} (%)	A_{rcw} (%)	A_{cw} (%)	A_{rcw} (%)
TBC-0.5	0.5	3.6	4.5	4.3	7.3
TBC-0.6	0.6	3.2	3.0	3.3	5.0
TBC-0.7	0.7	2.9	2.9	3.0	3.4

Results

Figure 3.32 shows that α^{act} values are large when $\Delta_{delay}(Y_{rcw})$ or $\Delta_{delay}(Y_{cw})$ values are small. This validates our assumption that the different testcases have similar trends (i.e., α^{act} versus $\Delta_{delay}(Y_{rcw})$ and $\Delta_{delay}(Y_{cw})$) even though the testcases have different clock periods, gate counts and core areas. Note that we only repeat the experiments for three different netlists. Is it possible that there are other netlists which show different trends compared to that in Figure 3.32.

Table 3.16 shows the timing analysis results with $\gamma = 0.0$. By using our methods (TBC-0.5, TBC-0.6 and TBC-0.7), we can improve the WNS by $46ps$ to $125ps$ and TNS by up to

[42] It is possible that using different $\alpha^{opt}(Y_{rcw})$ and $\alpha^{opt}(Y_cw)$ can improve the benefits of our signoff methodology.

$68ns$. Meanwhile, the total number of paths with timing violations is reduced by 42% to 100%.

Table 3.17 shows the results of a similar experiment with $\gamma = 0.5$. The results show that for all testcases, the $|G_{TBC}|$ are relatively smaller compared to that in Table 3.16 where $\gamma = 0.0$. This is because the A_{cw} and A_{rcw} are larger for the same α when there are stronger correlations among variation sources..

Table 3.17 shows that $|G_{TBC}|$ for the TBC-0.5 configuration is zero for the $LEON3MP$ testcase. Thus, the TBC-0.5 configuration has no improvements compared to the CBC approach. Meanwhile, results in Table 3.17 show that by using TBC-0.6 and TBC-0.7, we can still reduce the WNS up to $101ps$ and TNS by up to $53ns$ and the total number of paths with timing violations is also reduced by 10% to 100%.

The *delay estimation error* in Tables 3.16 and 3.17 are defined as $\Delta d_j(Y) - 3\sigma_{path_j}$. Since the delay estimation errors in the tables are positive, it means that no TBC case underestimates the delay variation.

To fix the remaining timing violation paths, we have several options. First, we can upsize standard cells along critical paths to reduce path delay. Second, if the wire delay is large, we can insert buffers to break long wires into shorter ones so as to reduce wire delay. Note that both approaches will change the $\Delta_{delay}(Y_{rcw})$ or $\Delta_{delay}(Y_{cw})$. If the $\Delta_{delay}(Y_{rcw})$ or $\Delta_{delay}(Y_{cw})$ becomes larger than the corresponding A_{rcw} or A_{cw}, we can use TBC, which will reduce the delay variation and improve WNS. Alternatively, we can also intentionally route the wires over multiple layers during the physical implementation stages so as to create critical paths which has less BEOL variations as already discussed in [166] [185].

Table 3.16: Timing analysis results with $\gamma = 0.0$.

	LEON3MP				*NETCARD*				*SUPERBLUE12*			
	CBC	TBC-0.5	TBC-0.6	TBC-0.7	CBC	TBC-0.5	TBC-0.6	TBC-0.7	CBC	TBC-0.5	TBC-0.6	TBC-0.7
WNS (ns)	-0.046	0.000	0.000	-0.010	-0.134	-0.009	-0.033	-0.059	-0.154	-0.085	-0.091	-0.106
TNS (ns)	-2.519	0.000	0.000	-0.043	-7.290	-0.030	-0.409	-0.894	-80.351	-18.899	-24.373	-34.993
#Timing violations	170	0	0	12	246	10	19	19	1422	869	972	1206
Delay estimation error (ns)	0.001	0.008	0.010	0.007	0.006	0.005	0.011	0.016	-0.001	0.006	0.003	0.007
$\|G_{TBC}\|$/total number of paths (%)	0.0	26.1	27.9	29.6	0.0	41.4	54.5	63.2	0.0	32.6	41.4	44.0

3.3.4 Conclusions

Due to highly resistive BEOL layers in advance technology nodes, signoff using conventional BEOL corners (CBC) results in longer chip implementation schedules and poorer design quality. We propose a method to reduce the pessimism in the CBC by using TBC. Our method

Table 3.17: Timing analysis results with $\gamma = 0.5$.

	LEON3MP				*NETCARD*				*SUPERBLUE12*			
	CBC	TBC-0.5	TBC-0.6	TBC-0.7	CBC	TBC-0.5	TBC-0.6	TBC-0.7	CBC	TBC-0.5	TBC-0.6	TBC-0.7
WNS (ns)	-0.046	-0.046	0.000	-0.010	-0.134	-0.134	-0.033	-0.059	-0.154	-0.146	-0.091	-0.106
TNS (ns)	-2.519	-2.519	0.000	-0.043	-7.290	-1.986	-0.434	-0.894	-80.351	-60.186	-27.039	-36.337
#Timing violations	170	170	0	12	246	35	20	19	1422	1229	1078	1276
Delay estimation error (ns)	0.001	0.000	0.011	0.010	0.005	0.004	0.011	0.019	0.000	0.002	0.006	0.002
$\|G_{TBC}\|$/total number of paths (%)	0.0	0.0	25.4	28.6	0.0	25.4	47.2	56.7	0.0	9.7	32.3	37.8

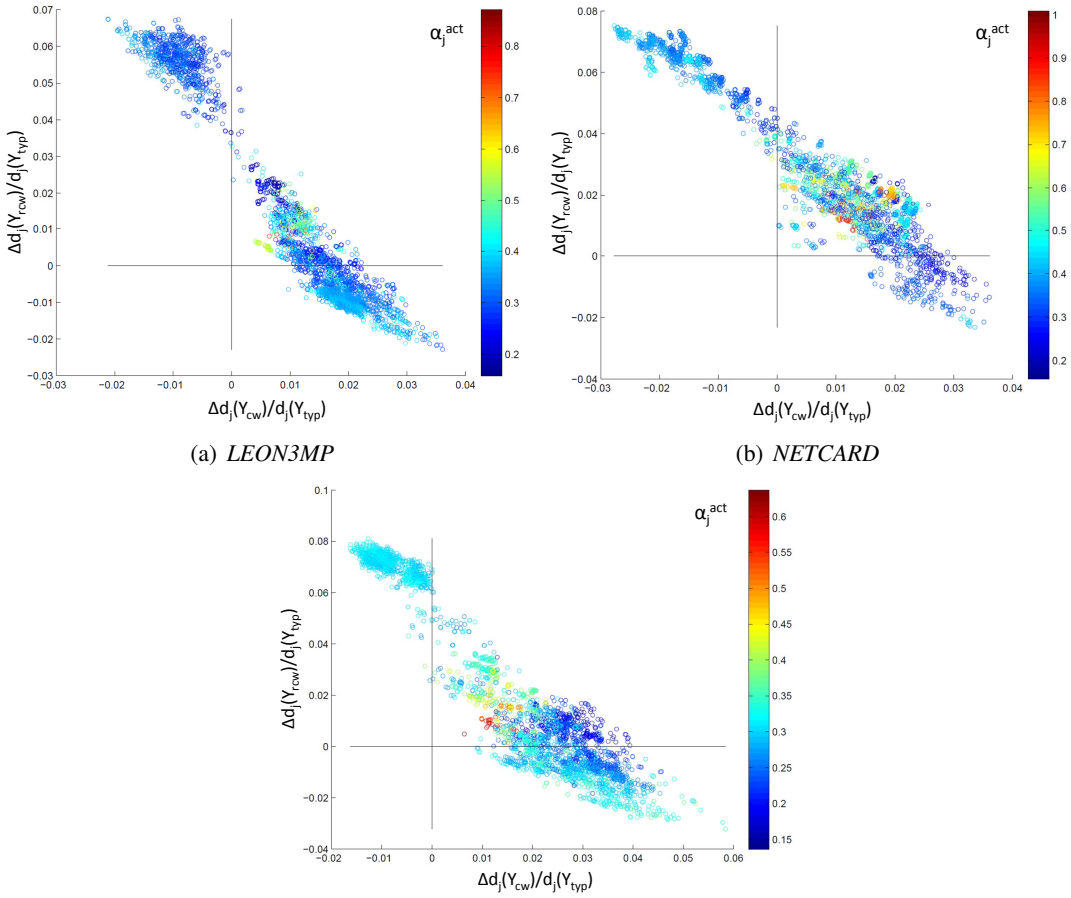
is based on the observation that most timing-critical paths use different BEOL layers. When the variations of BEOL layers are not fully correlated, the BEOL-induced timing variation is much smaller due to averaging of random variations.

Further, our analysis shows that by extracting the delay sensitivities of the critical paths to the RC-worst and C-worst BEOL corners, we can identify the paths which can use TBC for signoff without underestimating the delay variation (compared to a statistical analysis). The advantage of our method is that the TBC can be precharacterized and calibrated with statistical analysis when there is a major change in the technology node or design flow. Our experimental results show that our method which uses tightened BEOL corners on selected paths can reduce the number of paths with timing violations by up to 100% and improve the WNS and TNS by up to $101ps$ and $53ns$, respectively.

We observe that when the value of α is large the delay variations at Y_{cw} and Y_{rcw} are small. Thus, it may be possible to cover all critical paths by using a Y_{typ} with a small derating factor on wire delay. In other words, the design can be implemented and signed off by using Y_{rc_α}, Y_{rcw_α} and Y_{typ} (with a derating factor). We expect that this approach will further reduce the pessimism in BEOL corners because the design is not implemented at CBC.

3.4 Acknowledgments

Chapter 3 is in part a reprint of "On Aging-Aware Signoff with Adaptive Voltage", *IEEE Transactions On Circuits and Systems*, (to appear), "Impact of Adaptive Voltage Scaling on Aging-Aware Signoff", *Proc. Design, Automation and Test in Europe*, 2013, "Optimization of Overdrive Signoff in High-Performance and Lower-Power ICs", *IEEE Transactions on Very Large Scale Integration Systems*, (to appear), "Optimization of Overdrive Signoff", *Proc. Asia and South Pacific Design Automation Conference*, 2013, and "Timing Signoff Using Tightened Back-End-of-Line Corners in Advanced Technology Nodes", *Proc. IEEE International Confer-*



(a) *LEON3MP* (b) *NETCARD*

(c) *SUPERBLUE12*

Figure 3.32: Factor α^{act} versus $\Delta d_j(Y)$ of critical paths of different testcases.

ence on Computer Design, (to appear).

I would like to thank my co-authors Wei-Ting Jonas Chan, Sorin Dobre, Andrew B. Kahng, Jiajia Li, Siddhartha Nath and Bong-Il Park.

Chapter 4

Design-Aware Manufacturing Optimization

This chapter presents three distinct techniques for manufacturing optimization. First, we introduce a method to calculate the *electrical process window* (EPW) of a design which accounts for electrical specifications. The EPW is more accurate and less pessimistic compared to the conventional *geometric process window*, which only considers CD variation. We analyze various layout-transparent methods to enlarge the EPW to improve manufacturing yield. We also propose approximate methods to evaluate the EPW; these can be used with little or no design information. Furthermore, we propose a method to extract *representative layouts* for large designs which can then be used to evaluate the EPW with much smaller runtime. Second, we propose a design-dependent process monitoring strategy which can predict design performance based on measurements obtained from test structures in wafer scribelines. Since these measurements are available in the early stages of manufacturing, we propose to use the predicted design performance to prune bad wafers. Such early pruning can save test and back-end manufacturing costs. Third, we study the impact on BEOL electrical performance of stitching locations in LELE double-patterning mask design. We derive analytical RC equations to model the impact of CD variation due to the overlay error in LELE double patterning. Based on the analytical equations, we propose guidelines for optimal stitching to reduce RC variations.

4.1 Measurement and Optimization of Electrical Process Window

Process window (PW) is the range of process parameters such that designs produced within this range operate according to desired specifications [141]. The traditional geometric process window (GPW) checks whether the *critical dimension* (CD) of any feature deviates from its nominal value by more than a predefined tolerance [133] [141].

The rapid pace of semiconductor scaling over the last decades coupled with much slower advances in lithography technology has forced $193nm$ optical lithographic printing beyond its limit. Consequently, *resolution enhancement techniques* (RET) such as optical proximity correction (OPC), subresolution assist features and phase shift masks have become a necessity to ensure the printability of small features. Since OPC is typically performed at a nominal lithographic setup, it fails to account for variation in exposure, focus or overlay. To compensate for these variations, Krasnoperova et al. [116] propose a *process-window OPC*, in which OPCs are performed at multiple process corners. This method is, however, impractical due to its long runtime. Another method, *image slope OPC* [54] optimizes slope of intensity, which is a measure of variation in dose, along with *edge placement error* (EPE). Retargeting [180] [212] is a rule-based technique to modify the layout before performing OPC to improve process window and is a popular approach in industry. Although these methods address the problem of lithographic variation, accurate metrics are required to quantify their benefits.

Although GPW is easy to compute or measure, it is not an accurate representation of the electrical behavior of a printed circuit. Recently, there has been some interest in reducing the pessimism due to poor correlation between design geometry and electrical performance. In [9], electrically-driven OPC is developed based on nonrectangular transistor models for I_{on} and I_{off}. Zhang et al. in [216] developed an analytical model to account for corner rounding in printed transistors and accounted for its impact on saturation current during OPC. Gupta et al. in [86] used timing slack of critical paths to reduce the complexity of post-OPC mask shapes. These methods achieve smaller performance variation and reduced mask complexity despite large geometric errors [176]. Axelrad et al. [8] propose a methodology to compare the *static noise margin* (SNM) of 6T-SRAM cells printed under different defocus conditions. The method provides important feedback for designers at an early design stage, which helps to reduce design and manufacturing costs.

Inspired by the above-mentioned approaches, we propose an electrical process window (EPW), which estimates PW based on delay, SNM and leakage deviation instead of variation in CD. We focus on a PW analysis for digital VLSI circuit which has a dense geometry pattern and

is susceptible to lithographic variation. To evaluate EPW, we generate post-OPC lithography contours of a given layout at different exposure, defocus and overlay process points. A process point is denoted by O_k. Then, we extract transistor shapes and their electrical performances using the model in [33]. Finally, EPW is defined by process points that yield lithography contours with acceptable electrical performances.

The key contributions of this section are as follows.

- In contrast to the conventional GPW, we propose electrical process window defined by delay, SNM and leakage power of a design. EPW can reduce the pessimism in process control requirements as its area is 1.5 to 8$\times$ larger than that of GPW.
- We demonstrate that EPW can be optimized by layout-transparent methods such as gate length biasing and V_{th} adjustment during manufacturing.
- We propose several approximations to EPW for cases where design information is incomplete.
- We present the concept of *representative layout extraction* which can be used to reduce EPW evaluation runtime.

We focus on analyzing the lithography process window for the poly layer because it usually is the critical layer which affects circuit performance. Moreover, lithographic variation on poly layer has strong correlation to electrical variation as it defines transistor gate length.

Geometric Process Window

Definition: GPW is defined as the range of process parameters such that deviation between the CD of a printed contour and a circuit layout on the poly layer is within predefined tolerance, i.e.,

$$O_k \in GPW \iff \text{lower bound of allowed CD deviation} \leq \text{CD} \leq \text{upper bound of allowed CD deviation}. \tag{4.1}$$

In our experiments, CD deviation is estimated based on an EPE histogram of all transistor segments. As illustrated in Figure 4.1, EPE is defined as the displacement between the printed contour and layout shape. Since EPE only measures the channel length deviation on one side of a *transistor channel*, two scenarios are considered.

1. Maximum EPE occurs at both edges of a transistor segment. Allowed EPE = $\pm 2\times$ maximum EPE (worst case).

2. Maximum EPE occurs at one edge of a transistor segment. We assume that the edge opposite to the maximum-EPE segment is not changed, and that allowed EPE = nominal channel length $\pm$ maximum EPE.

We consider a process point O_k to be within GPW if more than 99% of EPEs are within the predefined tolerance. The 1% allowance is given to avoid pessimistic GPW due to EPE outliers, which can be fixed by fine-tuning the mask in OPC. In the following discussion, we use W-GPW and A-GPW to respectively denote GPW with the EPE tolerance defined by *Scenario 1* (worst case) and *Scenario 2* (average case).

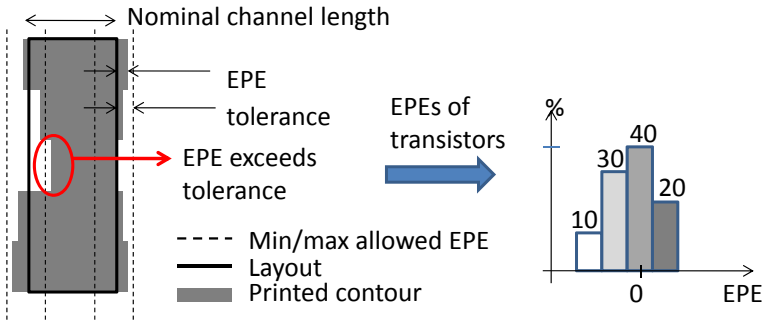


Figure 4.1: Illustration of EPE histogram.

4.1.1 Electrical Process Window

A process point O_k is considered within EPW if electrical metric of a printed circuit is within desired tolerance. In the following discussions, we demonstrate the evaluation of *delay-centric EPW* (D-EPW), *leakage power-centric EPW* (P-EPW) and *static noise margin EPW* (SNM-EPW) as they are commonly used electrical metrics.

Delay-Centric Electrical Process Window (D-EPW)

Due to subwavelength lithography, a printed transistor channel is not rectangular despite the use of aggressive RET techniques. This imposes difficulties in EPW extraction as electrical metrics of a *nonrectangular gate* (NRG) transistor cannot be determined from a precharacterized library. To model the impact of NRG transistors on a given critical path's delay, we extract I_{on} of each NRG transistor using the method proposed in [33]. As shown in Figure 4.2, NRG

transistor obtained from simulated contour is sliced into narrower transistors to approximate the nonrectangular channel. Then, the effective channel length, width and V_{th} of sliced transistors are extracted to construct rectangular transistors that correspond to the sliced transistors.[43] Finally, the rectangular transistors are simulated using *Synopsys HSPICE* [251] and their I_{on} and I_{off} are summed up to represent total I_{on} and I_{off} of the NRG transistor. After obtaining the current, cell delay of NRG transistor (d^{cell}) is estimated by the following equation,

$$d_i^{cell} = \frac{\sum_{n=1}^{N_{tran\_i}} I_{on\_ori\_n}}{\sum_{n=1}^{N_{tran\_i}} I_{on\_sim\_n}} \times d_{ori\_i}^{cell} \tag{4.2}$$

where N_{tran_i} is the total number of transistors in cell i and d_{ori}^{cell} is the delay of the cell obtained from STA such as *Synopsys PrimeTime* [254]. Subsequently, path delay of simulated contour (d_{sim}^{path}) is represented as the sum of delay of every cell along the path,

$$d_{sim\_j}^{path} = \sum_{i=1}^{N_{cell\_j}} d_i^{cell} \tag{4.3}$$

where N_{cell_j} is the total number of cells along the j^{th} critical path. Finally, D-EPW is defined as

$$\begin{aligned} O_k \in \text{D-EPW} &\Longleftrightarrow \max_j(\Delta d_j^{path}) \leq \text{upper bound of allowed delay deviation.} \\ \Delta d_j^{path} &= [\frac{d_{sim\_j}^{path}}{d_{ori\_j}^{path}} - 1] \times 100\%, \end{aligned} \tag{4.4}$$

where $d_{ori\_j}^{path}$ is the delay of the critical path obtained from STA.

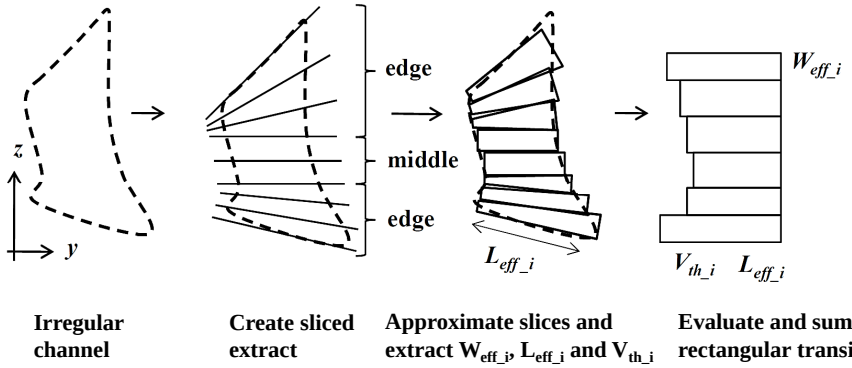


Figure 4.2: Non-rectangular gate transistor I_{on} and I_{off} extraction.

[43] We use SPICE-based method in [33] to calibrate parameters for NRG transistor model.

Leakage Power-Centric Electrical Process Window (P-EPW)

As already mentioned, leakage currents of NRG transistors at different process points (I_{off_sim}) are obtained using the method in [33]. The method is also used for calculating the leakage current of each transistor in pre-OPC layout (I_{off_ori}) to evaluate leakage power deviation of a circuit (ΔP).

$$\Delta P = [\frac{\sum_{n=1}^{N_{tran\_all}} I_{off\_sim\_n}}{\sum_{n=1}^{N_{tran\_all}} I_{off\_ori\_n}} - 1] \times 100\%, \tag{4.5}$$

where N_{tran_all} denotes the total number of transistors in a design. Note that Equation (4.5) does not account for cell topology, i.e., stacked transistors have less leakage power compared to non-stacked transistors. This leads to an estimation error whenever CD variations are different between the stacked and non-stacked transistors. Since the P-EPW is a function of relative leakage power instead of the absolute value, the estimation error is negligible if stack and non-stack transistors have similar CD distributions. For random digital logic, CD variation is affected by the surrounding pattern which has no direct correlation with its cell topology. Therefore, cell topology is unlikely a major source of estimation error.

Since there is no lower bound for leakage power, P-EPW is defined as

$$O_k \in \text{P-EPW} \Longleftrightarrow \Delta power \leq \text{upper bound of allowed leakage power deviation.} \tag{4.6}$$

Signal Noise Margin Electrical Process Window (SNM-EPW)

To capture the impact of lithography imperfection on a SRAM cell, we replace each NRG transistor in the cell by an *equivalent transistor* which has the same I_{on} as that of the NRG transistor. Since there can be many width and length combinations for a given I_{on}, we choose the equivalent transistor which has a channel width equal to the average width of the NRG transistor.

After obtaining the equivalent transistors for a SRAM cell, we perform SPICE simulation to get the *voltage transfer curves* of inverter pairs in a SRAM cell. We evaluate only the *read noise margin* of the SRAM, since it is typically more critical compared to the *hold noise margin*. The SNM of a cell is defined by the diagonal length of the maximum square within the butterfly curves as shown in Figure 4.3. Due to the regular layout of a SRAM array, the printed contour of each cell is similar. Therefore, we evaluate SNM-EPW based on the SNM value of a

SRAM cell. SNM-EPW is defined as

$$O_k \in \text{SNM-EPW} \Longleftrightarrow \Delta SNM \geq \text{lower bound of allowed signal noise margin deviation,} \tag{4.7}$$
$$\Delta SNM = [\frac{SNM_{simulated}}{SNM_{original}} - 1] \times 100\%.$$

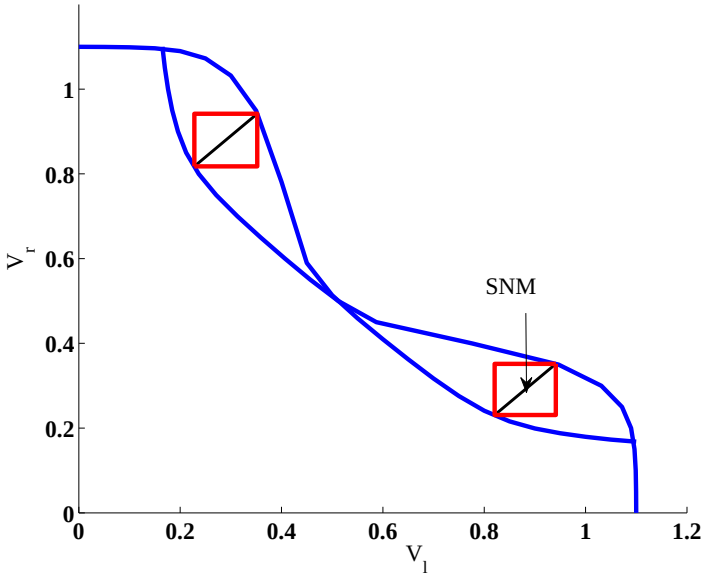


Figure 4.3: SNM extraction based on voltage transfer curves of a 6T-SRAM bitcell. V_r and V_l are the internal node voltage of inverter pairs in a bitcell.

Combined Electrical Process Window (C-EPW)

Whenever there are more than one electrical metrics, the combined electrical PW can be easily computed by finding the intersections of the EPWs,

$$\text{C-EPW} = \bigcap_{i=1}^{Q} (EPW_i), \tag{4.8}$$

where Q is the total number of electrical metrics. C-EPW is defined as the intersection between D-EPW and P-EPW.

Relationship Between GPW and EPW Tolerances

Since GPW and EPWs are defined based on different metrics, we need to figure out the relationship between the two for fair comparison. To obtain the worse case corners of GPW,

we simulate an inverter with a FO4 load and a 6T-SRAM cell at (nominal length $\pm$ (2 $\times$ EPE tolerance))[44] using *Synopsys HSPICE* [251] and transistor model provided by the *Nangate Open Cell Library* [240]. The maximum delay, leakage power and SNM deviations are extracted to represent D-EPW, P-EPW and SNM-EPW tolerances, respectively. Table 4.1 summarizes the corresponding deviations in delay and leakage power for different EPE tolerances. For example, a 5% EPE (2.5nm of 50nm nominal channel length) corresponds to 11%, 54% and -24% deviations in delay, power and SNM, respectively. Hence, W-GPW with 2.5% EPE tolerance corresponds to A-GPW with 5% EPE tolerance, D-EPW with 11% delay tolerance, P-EPW with 54% leakage power tolerance, and SNM-EPW with -24% SNM tolerance.

Table 4.1: Tolerances of GPW and EPW.

ΔChannel length (%)	W-GPW ΔEPE (%)	A-GPW ΔEPE (%)	D-EPW Δdelay (%)	P-EPW Δpower (%)	SNM-EPW ΔSNM (%)
5	2.5	5	11	54	-24
10	5.0	10	21	311	-61
15	7.5	15	30	2476	N/A

When channel length deviates more than 10%, the SNM of a 6T-SRAM cell reduces to zero. Therefore, the maximum allowed geometrical deviation is 10% for SRAM. The tolerance for leakage power is very high compared to channel length and EPE tolerance because leakage power increases exponentially as channel length reduces. Note that the tolerances in Table 4.1 are strongly dependent on the process technology.

Experimental Setup

To show the differences between GPW and EPW for digital logic, we implement six benchmark circuits obtained from *ISCAS-85* [230] and [243]. The benchmark circuits are implemented using the 45nm *Nangate Open Cell Library* [240]. After synthesis, placement and routing, we define the paths within 20% of setup time constraint as critical paths. The layouts of benchmark circuits are then scaled to 65nm for OPC and lithography simulation due to limitations in our optical models. After that, the simulated contours are scaled down to 45nm for leakage and drive current extraction. To emulate variations in the lithography system, we simulate an image for the poly layer with different exposure and defocus values using *Mentor Calibre* [228]. We only analyze the PW for the poly layer. During the EPW extraction, we use the active layer patterns in layout, i.e., we evaluate the PW for the poly layer when the active

[44] V_{dd} = 1.1V, tEmperature = 25°C

layer is printed at its nominal value. We emulate overlay error by shifting the printed active layer along the vertical direction (Z direction in Figure 4.2) during transistor shape extraction. Process parameters in our experiments are as follow.

- Exposure (%) $\in$ {80, 90, 100, 110, 120}
- Defocus (nm) $\in$ {0, 40, 80, 160}
- Overlay (nm) $\in$ {-20, -10, 0, 10, 20}

A process point is considered as "feasible" if all the transistors printed at this process point do not have open or short defects. The maximum process window is defined by the collection of all feasible process points. To evaluate GPW, we generate an EPE histogram for each process point by comparing the printed contours to the original layout using *Mentor Calibre* [228]. To evaluate EPW, we translate the extracted channel shapes into an *OpenAccess database* [241]. After that, I_{on} and I_{off} of each transistor are extracted using the method in [33] to obtain Δd_j^{path} and ΔP. Note that in order to reduce lithography simulation runtime, we estimate the delay, leakage power and EPE values between sampled data points by interpolation. The analysis of EPW (including NRG transistor current extraction) is implemented in C++ and the experiment is carried out on a 64bit machine running at 2*GHz* with 16*GB* memory.

Experimental Results

Results in Table 4.2 show that W-GPW is very pessimistic because its PW is zero for all tolerances. Meanwhile, A-GPW is larger than W-GPW because A-GPW has a a less constrainted CD tolerance. Figure 4.4 shows the A-GPW, D-EPW, P-EPW and C-EPW for benchmark circuit *C1908*.[45] Although the experiments are carried out for different exposure, defocus and overlay conditions, we do not show the PW along the overlay dimension because we observe that the PW is insensitive to overlay for the layouts we have. The experiment results for other circuits are not displayed but the area of the PWs are stated in Table 4.2. Figure 4.4[46] shows that the A-GPW is smaller than the EPWs with their corresponding tolerance. This implies that there are process points where the printed circuits can meet the electrical tolerance although the CDs of circuits violate geometric tolerance. GPW is generally more pessimistic compared to EPW because

[45]The result of W-GPW is not included in Figure 4.4 as it has zero area in all cases.

[46]Due to imperfect calibration of our OPC setup, the ideal process point at 100% exposure and $0nm$ defocus lies outside P-EPW at 54% tolerance, while the process points at 90% exposure and $0nm$ to $80nm$ defocus meets the tightest delay and leakage power tolerance.

1. GPW requires at least 99% EPE to be within CD tolerance. In contrast, EPW is defined based on the total power and delay of a circuit, which are related to the average of deviation of each transistor segment. Therefore, while some of the transistor segments can vary significantly, the entire transistor is still able to meet EPW tolerance due to averaging across transistors in a critical path for D-EPW, and across all transistors for P-EPW.

2. The transistors are not equally important in EPW, e.g., delay constraints are applied only to transistors on critical paths.

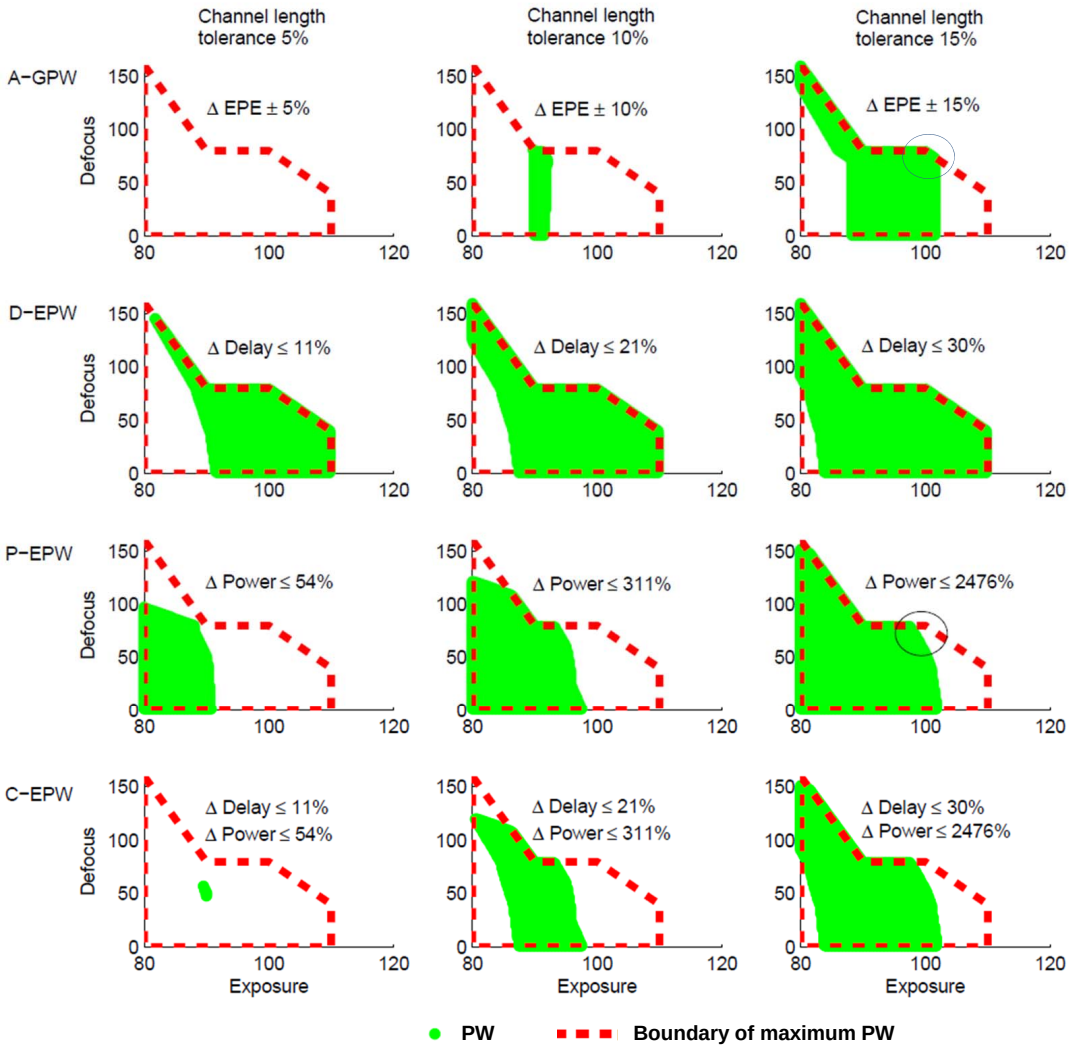


Figure 4.4: A-GPW, D-EPW, P-EPW and C-EPW for ISCAS-85 benchmark circuit *C1908*.

At the 100% exposure and 80nm defocus process point (circled in Figure 4.4), the geometric variation is within the A-GPW with $\pm 15\%$ EPE tolerance (shaded) but leakage power

Table 4.2: GPW and EPW for ISCAS-85 benchmark circuits.

	W-GPW			A-GPW			D-EPW			P-EPW			C-EPW (delay, power)			Maximum
Tolerance (%)	2.5	5	7.5	5	10	15	11	21	30	54	311	2476	(11, 54)	(21, 311)	(30, 2476)	PW
C432	0	0	0	0	300	1276	1538	2086	2460	882	1720	2107	0	1086	1846	2760
C499	0	0	0	0	117	1375	1559	2105	2508	921	1718	2076	9	1103	1864	2760
C880	0	0	0	0	196	1278	1390	1956	2332	825	1464	1969	0	890	1770	2565
C1355	0	0	0	0	95	1313	1665	2204	2560	847	1569	2052	35	1052	1891	2760
C1908	0	0	0	0	139	1253	1388	1937	2309	841	1493	1988	1	900	1767	2565
MIPS	0	0	0	0	0	190	921	1209	1426	334	599	823	0	248	690	1590
Average	0	0	0	0	141	1114	1410	1916	2266	775	1427	1836	7	880	1638	2500

deviation is not within the P-EPW with the corresponding tolerance. This happens when the actual channel-length deviation (combined EPE on both edges) is larger than $7.5nm$ (15% of channel length) but none of the EPEs exceeds $7.5nm$. As a result, the process point is valid for A-GPW but the actual leakage power is larger than the predefined leakage power constraints. This example shows that although A-GPW is generally pessimistic compared to EPW, it does not guarantee the electrical metrics of the circuits printed within its PW.

As shown in Figure 4.4, C-EPW can be much smaller than D-EPW or P-EPW. The C-EPW is useful as it clearly defines the acceptable process range, ensuring that the printed design can meet both delay and power requirements. For comparable tolerance, the C-EPW is $1.5\times$ to $8\times$ larger than A-GPW.

4.1.2 Optimization of Electrical Process Window

With EPW, the impact of process tuning on PW can be estimated from simulated contours. This enables fast and extensive exploration of process tuning approaches for maximizing PW. Since C-EPW is defined as the intersection of D-EPW and P-EPW, it is possible to improve C-EPW by increasing D-EPW or P-EPW. But any change in the gate lengths or V_{th} has opposite effects on D-EPW and P-EPW. For example, increasing the gate lengths of transistors leads to a larger P-EPW but a smaller D-EPW. This also implies that when the sensitivities of P-EPW and D-EPW to the intentional gate length or V_{th} perturbation are different, we can tune the gate length or V_{th} to improve C-EPW. We assume that $\pm 2nm$ gate length and $\pm 20mV$ V_{th} can be achieved through process tuning. To emulate the changes in gate length and/or V_{th}, we adjust the gate lengths and/or V_{th} of the transistors when we extract the I_{on} and I_{off}.

Figure 4.5 shows that reducing the gate lengths or lowering V_{th} enlarges D-EPW as

expected. Meanwhile, the P-EPW is reduced because the total leakage power is increased when the gate length or V_{th} is reduced. Since D-EPW only considers delay deviation on critical paths, reducing gate lengths on the critical cells (i.e., cells along the critical paths) or all cells has identical impact on D-EPW. For benchmark circuits *C880* and *MIPS*, however, this is not true because one or more of the reduced gate lengths on non-critical cells in the circuits are smaller than the minimum acceptable gate length ($30nm$). Any transistor smaller than this minimum gate length is considered to be electrically shorted and is a catastrophic circuit failure. As a result, the process points which print the shorted transistor are treated as not feasible points which reduce the D-EPW for circuit *C880* and *MIPS*.

Alternatively, we can improve P-EPW by increasing gate length or V_{th} of transistors. Figure 4.5 shows that increasing the gate length or V_{th} on (i) non-critical cells only or (ii) all transistors have similar improvements for P-EPW. However, increasing the gate length or V_{th} of all transistors reduces D-EPW because the delays on critical paths are also increased. For the testcases *C880* and *MIPS*, increasing gate lengths of non-critical cells have comparable impact to that of increasing gate lengths of all cells. This is because the number of critical cells is relatively small compared to the number of total cells as indicated in Table 4.3.

Table 4.3: Ratios of critical cells to total cells in benchmark circuits.

Circuits	Critical cells/total cells
C432	50%
C499	24%
C880	16%
C1355	49%
C1908	26%
MIPS	3%
Average	24%

On average, biasing gate lengths selectively increases C-EPW, while biasing gate lengths of all cells reduces C-EPW. Similarly, reducing V_{th} also increases C-EPW and vice-versa for incresing V_{th}.

4.1.3 Electrical Process Window Approximations

We propose two methods to estimate EPW using purely geometric means. This approach is useful when the information of the critical paths of a design is not available.

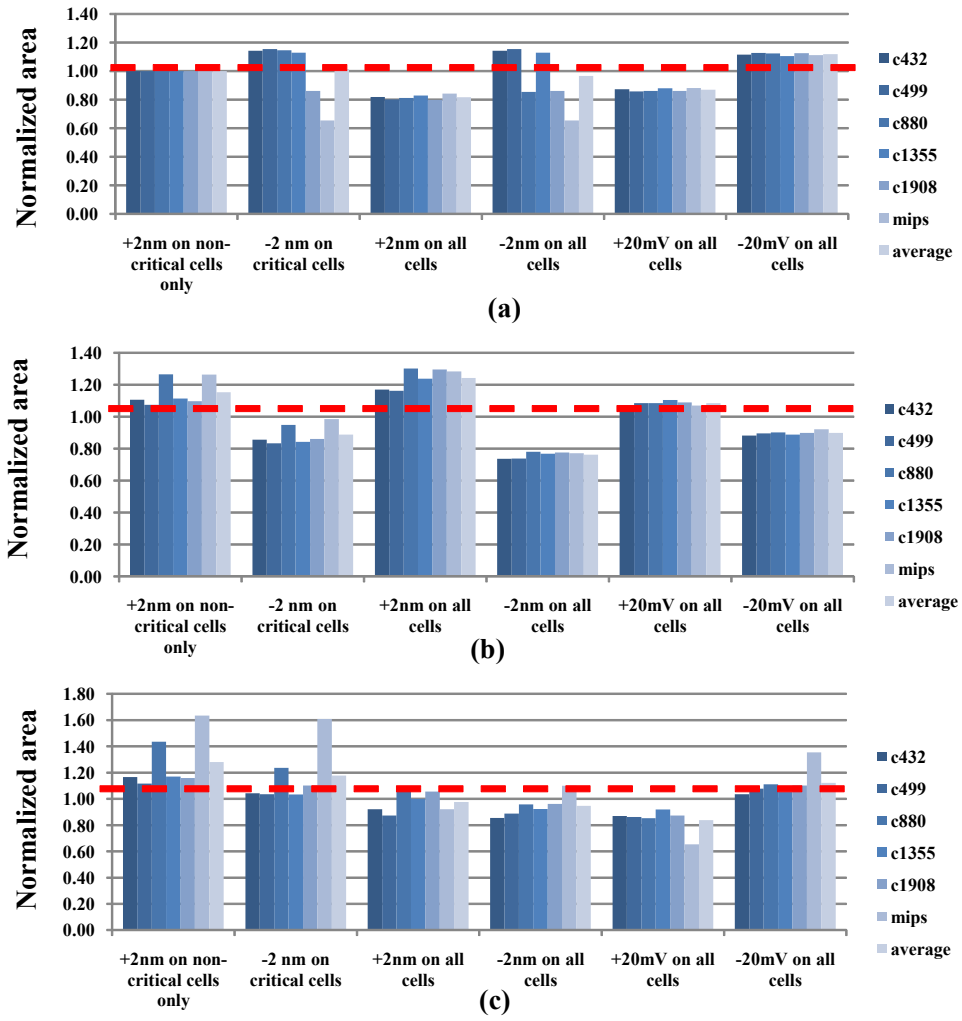


Figure 4.5: Optimized EPW area normalized to unoptimized EPW area for (a) D-EPW, (b) P-EPW and (c) C-EPW. Tolerances for delay and leakage power are 21% and 311%, respectively.

Method I: Use EPE Histogram of Entire Design

This method uses the EPE histogram generated during OPC to estimate an EPW without extracting the channel shape of each transistor. For a given design, we assume that the average delay and leakage power deviations induced by the EPEs of all transistors are approximately the same as that of an artificial equivalent transistor with the EPE histogram of the entire design. As illustrated in Figure 4.6, we translate each nonzero EPE bin into a transistor segment to create an equivalent transistor. Each transistor segment has the corresponding EPE in the histogram and the width of the segment is proportional to the percentage of the corresponding bin in the histogram.[47] Since the EPE can happen on both sides of a transistor, we define the channel length of the equivalent transistor as follows.

$$\text{channel length} = \text{nominal channel length} + 2{\cdot}EPE \tag{4.9}$$

After constructing the equivalent transistor, we can estimate its I_{on} and I_{off} by the NRG current extraction method mentioned in Section 4.1.1. Note that the EPE histogram is mainly constructed by the EPE of the middle part of transistor channels in a design. The middle part of a transistor have similar V_{th} as they are not affected by the *narrow width effects* which happens at the edges of a transistor. Therefore, we can ignore the narrow width effects in the equivalent transistor during the NRG current extraction, and the extracted current is independent of the ordering of transistor segments.

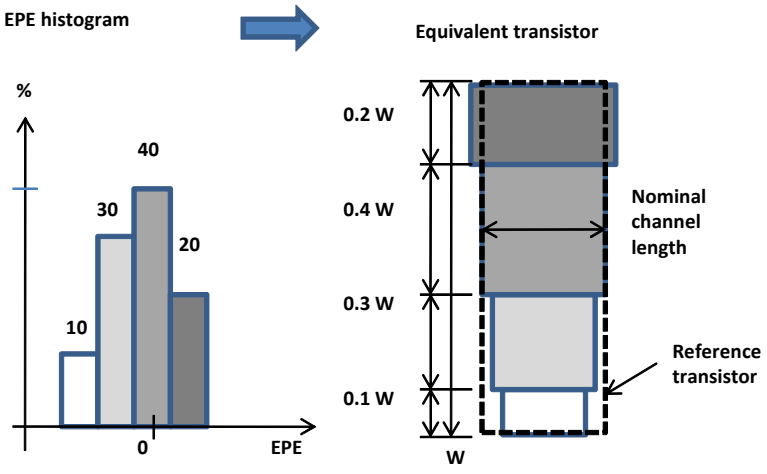


Figure 4.6: Extracting an equivalent transistor from the EPE histogram.

[47] If the segment in the EPE histogram have different width, the histogram can be weighed accordingly.

For the equivalent transistor, we estimate its delay deviation as the ratio of the I_{on} of a *reference transistor* (I_{on_ref}) to the I_{on} of the equivalent transistor (I_{on_equiv}). As shown in Figure 4.6, the reference transistor has a nominal channel length and its total channel width is the same as the width of the equivalent transistor. Meanwhile, we estimate the leakage power deviation of the equivalent transistor by the ratio of the I_{off} of the equivalent transistor (I_{off_equiv}) to the I_{off} of the reference transistor (I_{off_ref}). The histogram-based delay-centric EPW (histogram-D-EPW) and histogram-based power-centric EPW (histogram-P-EPW) are defined as follows.

$$\begin{aligned} &O_k \in \text{histogram-D-EPW} \Longleftrightarrow \\ &[\frac{I_{on\_ref}}{I_{on\_equiv}} - 1]\times 100\% \leq \text{upper bound of allowed delay deviation} \\ &O_k \in \text{histogram-P-EPW} \Longleftrightarrow \\ &[\frac{I_{off\_equiv}}{I_{off\_ref}} - 1]\times 100\% \leq \text{upper bound of allowed power deviation} \end{aligned} \tag{4.10}$$

In our experiments, an EPE histogram includes the EPE of PMOS and NMOS transistors. Since the widths of PMOS transistors are typically larger than the widths of NMOS transistors in CMOS circuits, we calculate the I_{on} and I_{off} of the equivalent transistor by the following equation.

$$I_{on} = \frac{\beta\times I_{on\_pmos} + I_{on\_nmos}}{\beta + 1} I_{off} = \frac{\beta\times I_{off\_pmos} + I_{off\_nmos}}{\beta + 1} \tag{4.11}$$

where β is the ratio of PMOS to NMOS channel width. In our experiments, we use the average β across different combinational cells in the *Nangate Open Cell library* [240] for the equivalent transistor ($\beta = 1.7$).

Method II: Use the Shape of Every Transistor

Given the shapes of all transistors, we can extract their I_{on} and I_{off}. Thus, we can calculate P-EPW based on the definitions in Equation (4.6) and no approximation is required. However, exact D-EPW cannot be determined because the information of critical cells is not available. Clearly, a strict D-EPW can be defined by the worst-case delay variation of all transistors. However, this definition is pessimistic since it ignores averaging effect along a critical path, which usually contains more than one single cell. To reduce the pessimism, we approximate D-EPW by averaging the N_{sample} largest delay variations. The delay variation of the n^{th}

transistor (Δd_n^{tran}) is given by

$$\Delta d_n^{tran} = [\frac{I_{on\_ori}}{I_{on\_sim}} - 1] \times 100\%, \tag{4.12}$$

where I_{on_ori} is the I_{on} of the pre-OPC transistor obtained from the layout and I_{on_sim} is the I_{on} of the NRG transistor from the simulated contour. The approximated D-EPW (shape-D-EPW) is defined as follows.

$$O_k \in \text{shape-D-EPW} \Longleftrightarrow \frac{\sum_{n=1}^{N_{sample}} \Delta d_n^{tran}}{N_{sample}} \leq \text{upper bound of allowed delay deviation} \tag{4.13}$$

In our experiments, we consider two N_{sample} for shape-D-EPW. First, we estimate N_{sample} based on the average number of gates on the critical paths in our benchmark circuits ($N_{sample} = 30$). Second, we assume that the EPE of transistors along a critical path is similar to that of all transistors in a design. Thus, N_{sample} = total number of transistors (N_{tran_all}).

Experimental Results

Figure 4.7 shows that the histogram-D-EPW is similar to the reference D-EPW. But the histogram-P-EPW is significantly smaller than the reference P-EPW. As a result, the approximated histogram-C-EPW only covers a small region of reference C-EPW. The error in histogram-P-EPW is mainly due to the definition of channel length in Equation (4.9), which considers the worst-case EPE scenario. Also, the error in channel length is more significant for P-EPW as leakage power grows exponentially when the channel length shrinks.

Figure 4.7 shows that shape-D-EPW and shape-C-EPW with $N_{sample} = 30$ is much smaller than that of reference EPWs. The accuracy of the approximation improves when N_{sample} is equal to N_{tran_all}. Since the evaluation of shape-P-EPW is the same as the one for reference P-EPW, there is no difference between them.

In Figure 4.8, we see that all approximation methods have larger EPWs compared to A-GPW (on average). When both leakage and delay are considered, the shape-C-EPW with N_{sample} = N_{tran_all} has the largest PW. The EPW of the shape-D-EPW with N_{sample} = N_{tran_all} is slightly less than the histogram-D-EPW although both approximations use the average delay deviation of all transistors to calculate D-EPW. This discrepancy is due to the difference between the EPE histogram and the actual transistor shape. Note that the histogram-D-EPW is larger than the reference EPW. This happens because histogram-D-EPW is evaluated based on the EPE histogram of the entire design, while D-EPW only considers the transistors along critical paths.

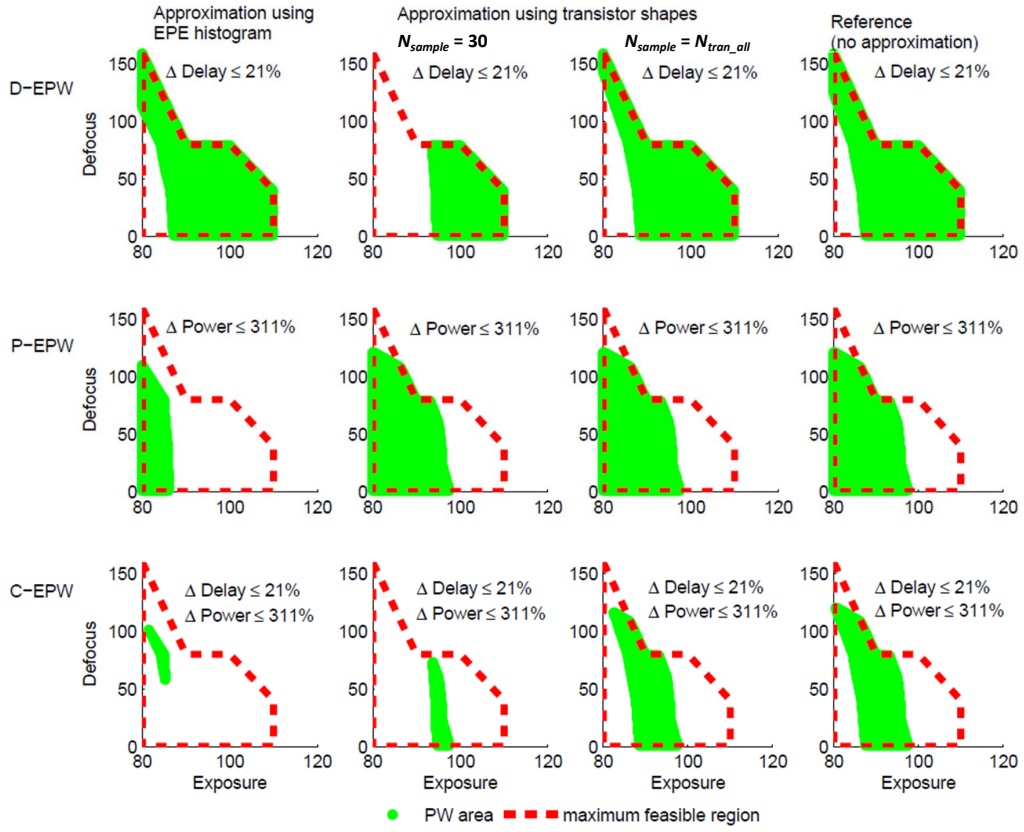


Figure 4.7: Comparison between EPW and its approximations for benchmark circuit *C1908*.

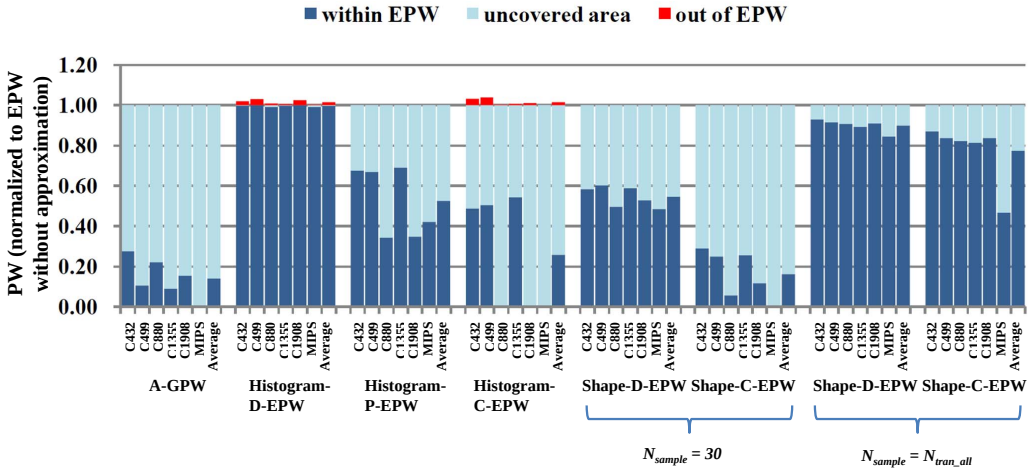


Figure 4.8: Accuracy analysis for A-GPW and approximated EPWs of benchmark circuits. EPE tolerance = 10%, delay tolerance = 21% and leakage power tolerance = 311%.

In summary, EPW extracted based on the shape of each transistor (with N_{sample} = N_{tran_all}) is the best approximation as it has no area out of EPW and covers $> 70\%$ of the reference EPW (on average).

Runtime Reduction Through Representative Layout Extraction

All the GPWs and EPWs mentioned above require lithography simulations of the shapes of a given design at multiple process points. The runtime for lithography simulations is very long especially for process points at a finer level of granularity.

To reduce the lithography simulation runtime, we propose an efficient PW analysis flow depicted in Figure 4.9. First, we extract representative layouts (RLs) which contain relevant shapes for EPW analysis. We select all standard cells along critical paths for D-EPW and 5% of the total cells in a design for P-EPW. Second, we check the printed image of the original layout for all process points and filter out the process points which have pinching/bridging (i.e., short circuit) features. This can be done efficiently by using a less accurate but fast lithography simulation setup.[48] In the case where the selected cells are too many for an efficient lithography simulation, we apply an additional clustering procedure to further reduce the total number of cells. Through the sampling and clustering techniques, the lithography simulation runtime is reduced because these RLs have smaller feature counts as compared to the original layouts.

Representative Layout Extraction

To estimate delay deviation, we only consider transistors on critical cells because they are more likely to cause a timing violation compared to other cells. To construct representative layouts for delay estimation, we take a $2\mu m \times 2\mu m$ square snippet centered at each transistor's channel (of each critical cell) to form basic layout snippets. The size of snippets is chosen to account for optical proximity effects on the transistor under consideration. After collecting all layout snippets, we tile the layout snippets to create a *delay representative layout* (DRL) for the design.

Since each transistor contributes to the total leakage power, there is no obvious selection scheme to extract "critical" shapes to estimate power deviation. Therefore, we sample 5% cell instances from each cell type. For each sampled cell, we take a $2\mu m \times 2\mu m$ snippet for each transistor in the cell. We then tile the layout snippets to construct a *power representative lay-*

[48] Note that identifying PW to avoid bad pinching/bridging patterns is not sufficient as there are patterns which can only tolerate small errors due to design-specific timing constraints.

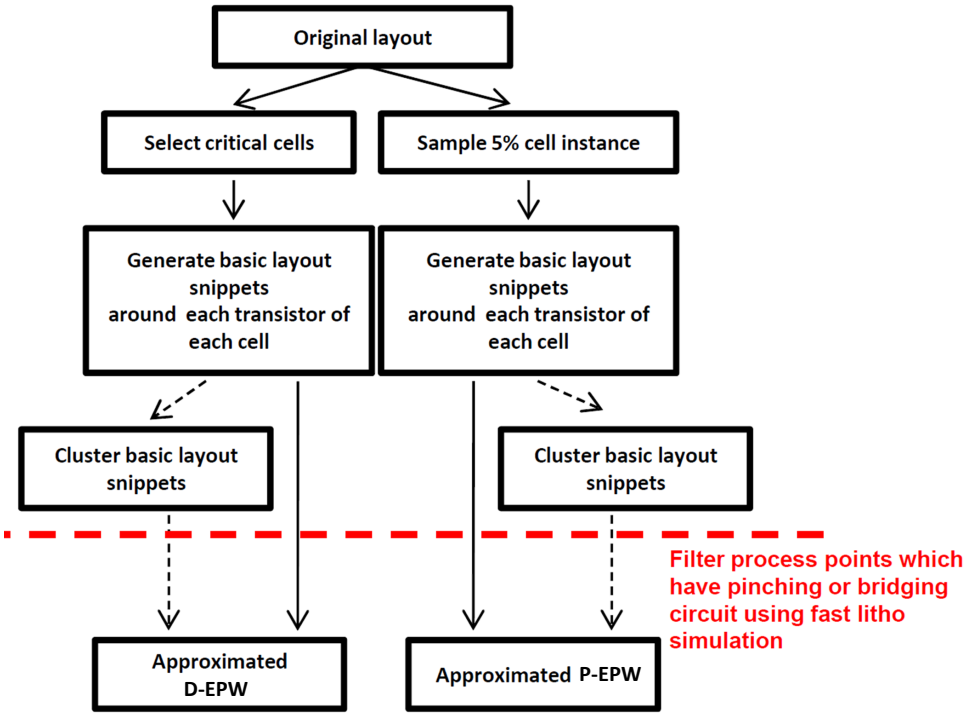


Figure 4.9: Clustering flow.

out (PRL). This sampling approach reduces runtime while minimizing estimation error because standard cells with the same cell type are likely to have similar leakage power deviation.

Only DRL and PRL of a design layout then undergo a lithography simulation at different process corners to evaluate EPW. Note that while we use neighboring shapes of a transistor during RL extraction, we only perform EPW analysis on the transistor in the middle of the snippet for both DRL and PRL. We apply the approximate EPW methods discussed earlier to the representative layouts because complete EPW analysis is not applicable due to the lack of information of critical paths. Table 4.4 shows that the total lithography simulation runtime of two RLs is substantially less than that of the entire design layout for the *MIPS* testcase.[49]

We can further reduce total transistor shapes that need to undergo the lithography simulation by clustering the chosen layout snippets using the method in [80]. The runtime improvement due to clustering is also shown in Table 4.4.

Figure 4.10 shows the accuracy of our DRL+PRL extraction method compared to evaluation of EPW for the entire design. The results show that the PW estimated using representative layout method is similar to the one which uses entire design. The shape approximation method

[49]The runtime values are the CPU TIME as reported by *Mentor Calibre* [228].

Table 4.4: Lithography runtime for representative layouts.

Benchmark Circuit	Total cells	Critical cells	Lithography Runtime (Hours)		
			Full Design	Representative Layout	Post-clustering
MIPS	11577	382	198	101	93

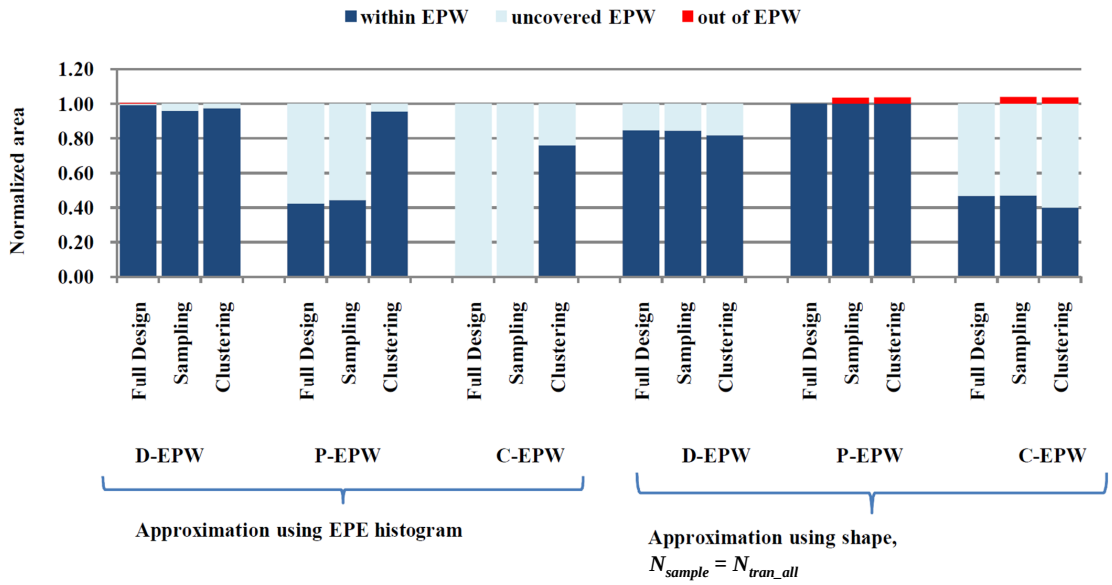


Figure 4.10: Accuracy of clustering approach for benchmark design *MIPS*.

is slightly optimistic as it overestimates P-EPW. This is because the random sampling misses out some critical patterns that cause leakage power failure. Note that there is no area out of EPW for the histogram method. This happens because the error in sampling is compensated by the pessimistic estimation of the histogram method. In summary, the RL extraction method reduces lithography simulation runtime significantly at the cost of EPW accuracy (i.e., the representative snippets do not have all the features of the critical geometries).

EPW Including SRAM

To evaluate the EPW of digital circuits, we need to consider the PW for random logic as well as memory cells. Since the original benchmark circuits do not have memory cells, we draw the layout of a SRAM according to the geometrical dimensions in [20]. After that we optimize the bitcell by upsizing the pull-down transistors from $80nm$ to $120nm$. This improves the static noise margin from $163mV$ to $213mV$. The area of the upsized bitcell is $2.9\mu m^2$ ($0.785\mu m \times 0.370\mu m$). In our experiments, we duplicate the layout of a 6T-SRAM cell to form a memory

array for lithography simulation. During the PW analysis, we evaluate the bitcell in the middle of the array, which is not affected by empty patterns around layout boundaries.

GPW versus EPW

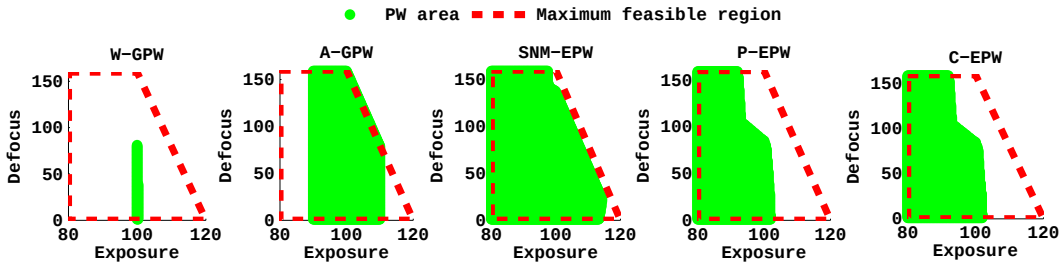


Figure 4.11: SRAM GPW versus EPW.

Figure 4.11 shows that SNM-EPW is much larger than GPW because SNM is affected by the relative "drive strength" of transistors instead of absolute critical dimension deviation. For example, when the channel length of all transistors increases due to lithographic variation, the impact of I_{on} reduction in the pull-down transistors is compensated by I_{on} reduction of the access transistors. As a result, the SNM of a SRAM cell may still lie within tolerance even though the printed contour violates geometrical tolerance.

To perform a full EPW analysis on benchmark circuits, we define C-EPW as the intersection of delay, power and SNM-EPW. We use $\pm 10\%$ CD tolerance for SRAM and $\pm 10\%$ CD tolerance for random logic in our experiments.

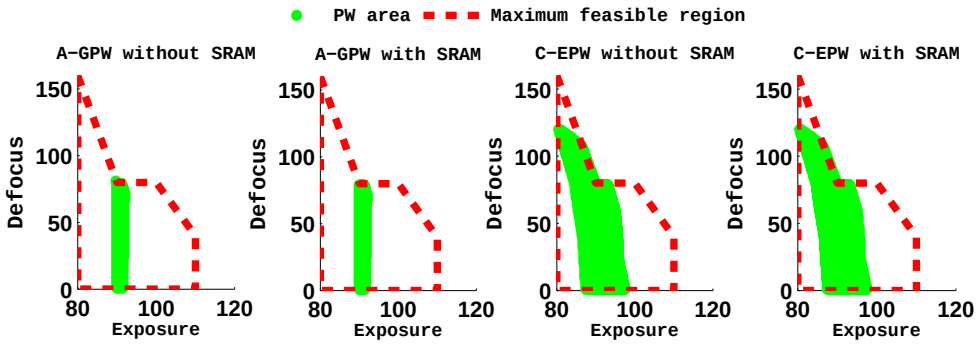


Figure 4.12: GPW versus EPW for benchmark circuit *C1908*.

Figure 4.12 shows that both GPW and C-EPW do not change after intersecting the digital logic and SRAM PWs. This implies that the SRAM bitcell is not a limiting factor for PW. The

results in Table 4.5 show that C-EPW is about 8× larger than GPW on average for digital logic and SRAM circuits.

Table 4.5: GPW and EPW areas with SRAM.

	A-GPW	C-EPW (delay, power, SNM)	Maximum PW
C432	300	1086	2760
C499	117	1103	2760
C880	196	890	2565
C1355	95	1052	2760
C1908	139	900	2565
MIPS	0	248	1590
Average	109	839	2448

Impact of SRAM on Approximation Methods

We also study the impact of including SNM-EPW to the approximation methods in Section 4.1.3. Figure 4.13 shows that the C-EPWs (including SRAM C-EPW) of approximation methods are greater than the PW of GPW. Including SNM-EPW in the C-EPW does not change the result of approximation methods (see Section 4.1.3) because the SNM-EPW is not the limiting PW in this case. Similarly, Figure 4.14 shows that including SNM-EPW does not change the results of our representative layout approaches.

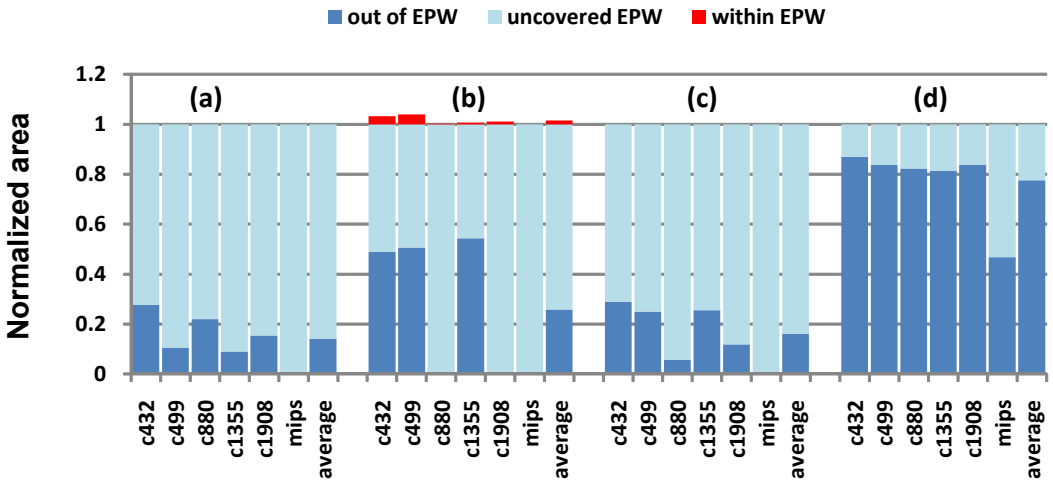


Figure 4.13: Accuracy of (a) A-GPW, (b) C-EPW using histogram approximation (c), C-EPW using shape approximation with N_{sample} = 30, and (d) C-EPW using shape approximation with $N_{sample} = N_{tran\_all}$. C-EPW includes SNM-EPW.

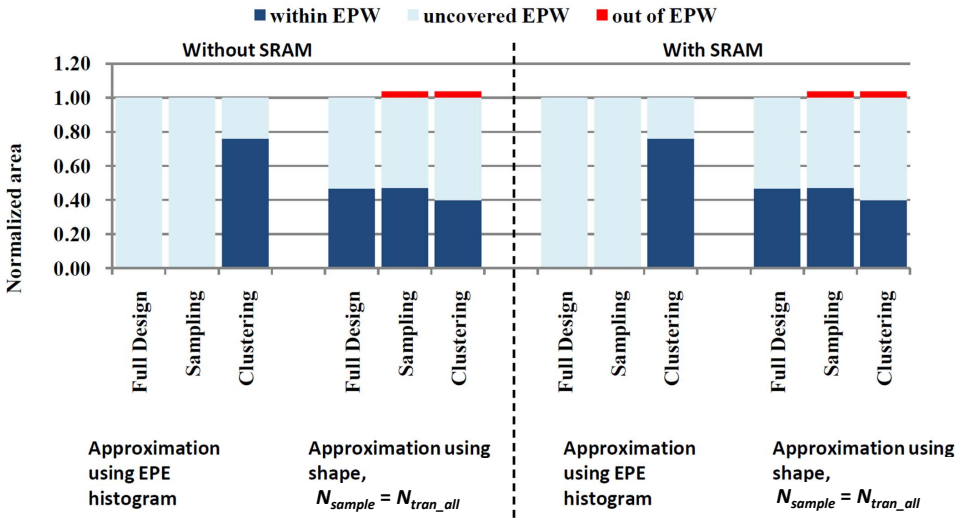


Figure 4.14: Accuracy of clustering approach including SNM-EPW for benchmark design *MIPS*.

4.1.4 Conclusions

We propose the concept of an electrical process window, which is a better measure of process window than the conventional geometric process window. Our experimental results show that the area of EPW is 1.5× to 8× larger than the GPW because it removes the inherent pessimism of GPW by averaging the impact of geometric variation on electrical parameters. We also analyze various layout-transparent methods to enlarge EPW. Based on our experiment results, we find that gate-length biasing and V_{th} adjustment can improve EPW by approximately 10%.

The calculation of delay-centric EPW requires information of critical cells in design which is often not available to foundries. Hence, we propose two approximations to EPW, one based on EPE histogram and the other based on transistor shape analysis. Our results show that the EPW approximated by using the transistor shape covers more than 70% of the reference EPW on average. We also propose a method to extract representative layouts which can be used to reduce the runtime in process window calculation by 49%. Though we demonstrate the process window analysis under defocus and exposure variations, other lithographic nonidealities such as mask error can be included in the lithography simulation.

4.2 Design Dependent Process Monitoring

Process variation has been a critical aspect of semiconductor manufacturing [117]. When new process technologies are introduced, process variation causes manufactured chips to exhibit a wide performance spread [21], and wafer yield could be as low as 30% to 50% [207]. Although screening defective chips after manufacturing can reduce burn-in, testing, and packaging costs [179], the chips until this point have already incurred unnecessary manufacturing cost. Thus, it is beneficial to prune bad wafers and chips during the early stages of manufacturing wherever possible using low-cost tests.

Early wafer pruning has been introduced in [145], where *cost-of-yield* (COY) is defined as a metric to guide the decision of pruning or scrapping a wafer in production. Based on a comprehensive cost analysis on wafer pruning, Wu et al. [207] propose a genetic algorithm for making a wafer lot pruning decision. These wafer pruning strategies do not address the problem of estimating chip performance and consequent parametric yield at the early wafer manufacturing stages for wafer-level pruning.

Mitra et al. in [151] show an example of early chip-performance estimation by using RO delay as a measure of chip performance. This method relies on the correlation between RO and the chip's critical paths, which is inherently inaccurate as every critical path has a different sensitivity to process variation. Since inaccurate chip performance estimations may lead to wrong pruning decisions, it is necessary to have an accurate design-dependent process monitoring method. Meanwhile, the monitoring structures should be placed in the wafer scribeline to minimize the measurement cost and silicon area overhead. Though RO-guided testing strategies are common [30] [110], we have not seen any previous work dealing with designing scribeline ROs which are design-specific.

To capture design-specific performance variation, Liu and Sapatnekar [137] propose a framework to estimate chip performance with post-silicon measurement. This method assumes that the distributions of process variations as well as the correlation among the variation sources are given. Cho et al. in [53] propose to train a neural network for chip performance prediction by using the data collected during manufacturing. The accuracy of the estimation is strongly related to the training data. For both methods, the required process information and training data are usually not available or inaccurate as process parameters are varying.

Design-specific monitors have been proposed in [65] [137] [178]. However, these monitors are not suitable for low-cost scribeline-based test for several reasons. Scribeline test structures are designed and tested by the foundry using a probe card; using customized test structures

and testing procedures will increase cost and manufacturing complexity. Also, the monitoring circuits may be too large to fit into the scribeline, which has a limited area. Another disadvantage of using on-chip monitors (e.g., [65] [137] [178]) is that probing on-chip monitors at an early manufacturing step will introduce defective particles around the monitor, which will reduce the wafer yield. Using scribeline structures poses a lower risk of introducing defective particles because probing is not directly applied on the chip.

In Section 4.2, we propose a design-dependent monitoring approach using commonly used compact scribeline test structures (e.g., those in [131]). These test structures are generic and capable of measuring the following parameters after the M1 stage of manufacturing.[50]

$$\begin{aligned} I_h &= I_{ds} \quad \text{at } V_{gs} = V_{dd}, \quad V_{ds} = V_{dd}/2 \\ I_l &= I_{ds} \quad \text{at } V_{gs} = V_{dd}/2, \quad V_{ds} = V_{dd} \\ I_{off} &= I_{ds} \quad \text{at } V_{gs} = 0, \quad V_{ds} = V_{dd} \\ C_{gate} & \quad \text{at } V_{gs} = V_{dd}, \quad V_d = V_s = 0 \end{aligned} \tag{4.14}$$

where V_{dd}, V_{gs}, V_{ds}, V_d and V_s are supply, gate-to-source, drain-to-source, drain and source voltages, respectively. I_h, I_l and I_{off} are drain-to-source current (I_{ds}) of a CMOS device (NMOS or PMOS) at the corresponding bias conditions. C_{gate} is gate capacitance of a device. Based on the measured values of I_h and I_l, we can represent circuit delay with effective drive current (I_{eff}), defined as [154]

$$I_{eff} = \frac{I_h + I_l}{2} \tag{4.15}$$

At the early stage of wafer manufacturing, we estimate the delay and leakage power of a chip by using the I_{eff} and I_{off} measured from test structures. Based on the estimated timing and leakage power, a wafer and chip pruning decision can be made for manufacturing cost reduction. The overview of our method is depicted in Figure 4.15.

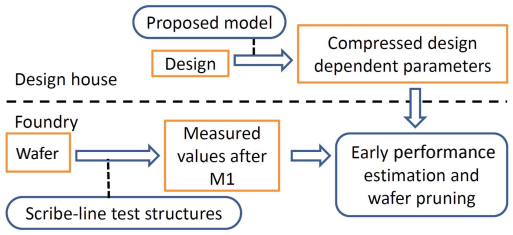


Figure 4.15: Overview of wafer and chip pruning methodology.

[50]The bias points match commonly used measurements on the scribeline process control monitoring test circuits in commercial foundries.

Our contributions are as follows.

- We propose a scribeline-based design-dependent approach for chip performance and leakage power estimations.
- We analyze the within-die variation and measurement noise effects in the chip performance estimations.
- We show how the above information can be used to accurately identify bad wafers and help in wafer pruning and yield estimation.
- Using the estimated chip delays, we show that bad dies can be readily identified and pruned from the testing lot, to save on costly tester time.

4.2.1 Delay Estimation Using I_{eff}

We model chip delay using I_{eff}, which is defined as the average current that charges or discharges a circuit node during a logic transition.[51] The delay of a logic transition is modeled as

$$delay \propto \frac{CV}{I_{eff}} \tag{4.16}$$

where C is the node capacitance, V is the voltage swing and I_{eff} is the effective drive current. While I_{eff} cannot be physically measured, several works propose approximations using device level *I-V* characteristics [6] [87] [154]. Though more complex models (e.g. [6]) can be used as well, our experiments indicate that Equation (4.15) suffices for our device models and libraries.

Cell Delay Model

Using Equation (4.16), we express the delay of the i^{th} cell as

$$d_i^{cell}(c) = \sum_{t \in T} \frac{K_i^{cell}(c,t) \cdot C_i \cdot V}{I_{eff}(t)} \tag{4.17}$$

where $K_i^{cell}(c,t)$ is the delay scaling coefficient for the i^{th} cell, c denotes the cell type (e.g., INV, NAND, etc.), t denotes device type, T is the set of all device types and C_i is node capacitance of

[51] If scribeline measurements for electrical parameters such as V_{th}, channel length, electron mobility, etc., are available, our delay model can be modified to incorporate the impact of these parameters to improve delay estimation.

the i^{th} cell.[52] $K_i^{cell}(c,t)$ is fitted for different input slew, output load and transition combinations. This fact is implicit and we do not show it for notational convenience.

Expanding d_i^{cell} using *Taylor series* with respect to $I_{eff}(t)$ for all $t \in T$ and ignoring the cubic and higher order terms, we get

$$d_i^{cell}(c) = d_{nom\_i}^{cell}(c) - \sum_{t \in T} \frac{K_i^{cell}(c,t) \cdot C_i \cdot V}{I_{eff\_nom}(t)} \left(\frac{\Delta I_{eff}(t)}{I_{eff\_nom}(t)} - \frac{\Delta I_{eff}^2(t)}{2 I_{eff\_nom}^2(t)} \right) \tag{4.18}$$

where $I_{eff\_nom}(t)$ is the nominal $I_{eff}(t)$ and $\Delta I_{eff}(t)$ is the $I_{eff}(t)$ change due to process variations. $d_{nom\_i}^{cell}$ is the nominal delay of the i^{th} cell. $K_i^{cell}(c,t)$ are fitted for every cell using (4.18) by varying process conditions for different input slew and output load points. This model fitting can be done very efficiently as it can use existing process specific timing libraries which are available for various corners. In our experiments, we do not have access to a sufficient number of these libraries. Therefore, we fit the model using SPICE simulations on individual cells.

Path Delay Model

The delay of the j^{th} path (p_j) under process variations can be expressed as

$$d_j^{path} = d_{nom\_j}^{path} + \Delta d_j^{path} \tag{4.19}$$

where $d_{nom\_j}^{path}$ refers to the nominal delay of p_j. Δd_j^{path} is the delay change due to process variation, which is equal to the sum of delay changes of every cell in the path,

$$\Delta d_j^{path} = - \sum_{i \in G_j} \sum_{t \in T} \frac{K_i^{cell}(t) \cdot C_i \cdot V}{I_{eff\_nom}(t)} \left(\frac{\Delta I_{eff}(t)}{I_{eff\_nom}(t)} - \frac{\Delta I_{eff}^2(t)}{2 I_{eff\_nom}^2(t)} \right) \tag{4.20}$$

where G_j is the set of cell instances on p_j. Due to process-induced variation on slew and load, K_i^{cell} may differ from its value extracted during the design time. To evaluate the process-induced variation on K_i^{cell}, we simulate standard cells with 1000 randomly sampled process conditions based on the variation model in Table 4.7 (see Section 4.2.4). We then extract the input slew and output capacitance of the standard cell and calculate its K_i^{cell} based on the proposed delay model. Results of this study show that standard deviation of K_i^{cell} (average of INV, NOR2 and NAND2 gates) is 6.0%. Although our model does not capture the process-induced K_i^{cell} variation, error

[52] We take four device types into account: {high V_{th}, low V_{th}}×{PMOS, NMOS}. Standard cells made by the same device type have two nonzero $K_i^{cell}(c,t)$ coefficients.

induced by K_i^{cell} variation is included in our experiments.

The sensitivity of delay of p_j to changes in $I_{eff}(t)$ can be expressed as[53]

$$K_j^{path}(t) = \sum_{i \in G_j} K_i^{cell}(t) C_i \tag{4.21}$$

The total path delay can now be written as

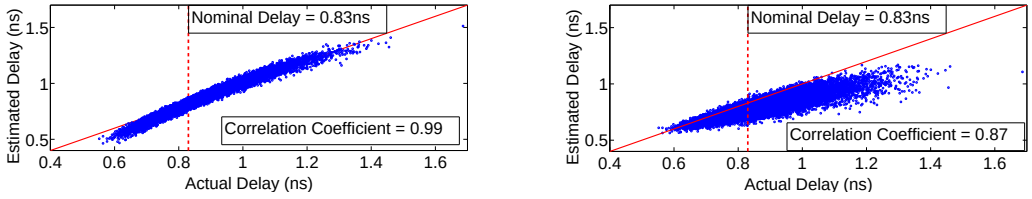
$$d_j^{path} = d_{nom\_j}^{path} - \sum_{t \in T} \frac{K_j^{path}(t) V}{I_{eff\_nom}(t)} \left(\frac{\Delta I_{eff}(t)}{I_{eff\_nom}(t)} - \frac{\Delta I_{eff}^2(t)}{2 I_{eff\_nom}^2(t)} \right) \tag{4.22}$$

Handling Load Capacitance Variation

In Equation (4.21), the path-specific delay sensitivities to I_{eff} depend on the nominal value of output load, which is seen by the cells. However, with process variations, this output load also changes. Therefore we scale the estimated delay by the ratio of the actual capacitance to its nominal value (C_{gate_nom}).

$$d_j^{path'} = (d_j^{path} - d_{interconnect\_j}^{path}) \frac{C_{gate}}{C_{gate\_nom}} + d_{interconnect\_j}^{path} \tag{4.23}$$

where $d_j^{path'}$ is the scaled delay estimation and $d_{interconnect\_j}^{path}$ is the interconnect delay of a path.[54]



(a) Proposed delay model (b) Design independent model

Figure 4.16: Delay estimated by (a) the proposed delay model, and (b) a design-independent approach, compared with actual delay for an *C432* benchmark, obtained from static timing analysis with timing tables characterized at the randomly sampled process conditions.

Figure 4.16 shows the accuracy of the proposed design-dependent delay estimation technique using (4.23), compared to a design-independent approach. In this experiment, we ran-

[53] $K_j^{path}(t)$ is instance-dependent as input slew and output load may vary with instance.

[54] Interconnect delays extracted from our benchmark designs are much smaller than cell delays. For simplicity, we scale the entire path delay by the ratio of actual device capacitance to nominal capacitance in our experiments.

domly generate 1000 process condition samples for the variation model in Section 4.2.4 (without within-die variation). We then characterize timing libraries for all standard cells (using SPICE) at the sampled process conditions to calculate the worst-case (actual) delay of the *C432* benchmark circuit using static timing analysis. Meanwhile, we extract I_{eff} and I_{off} of the PMOS and NMOS devices at the same process conditions using the SPICE simulator. After that, we apply Equation (4.23) to obtain delay estimations for the proposed delay model. Since a design-independent delay estimation has no information about the circuit, we assume that the design-independent approach equally weights all device types and calculate path delay as follows.

$$d_{indep\_j}^{path} = d_{nom\_j}^{path} \cdot \sum_{t \in T} \frac{I_{eff}(t)}{I_{eff}(t) + \Delta I_{eff}(t)} \tag{4.24}$$

where $d_{path\_indep\_j}$ is the path delay estimated by a design-independent approach. The result shows that the proposed delay estimation tracks the actual delay well. The correlation coefficient is found to be 0.99, compared to 0.87 for the design-independent approach. This is because the design-independent methodology is oblivious of the exact nature, topology and the structure of the cells that make up the critical paths in the design, while our strategy effectively captures this dependence in the K_j^{path} form.

Effect of Within-Die Variation on Delay

Inter-die variation is being captured by scribeline test structures available next to each die. However, measurements from test structures are typically different from the ones on critical paths due to within-die variation. We express the within-die variation as a normally distributed random variable with zero mean and standard deviation, $\mathcal{N}(0, \sigma_{wd})$. The distribution can be estimated by making multiple measurements per die.[55] Considering only the first order term in Equation (4.22), the path delay vector can be rewritten in matrix form as

$$\mathbf{D} = \begin{bmatrix} d_1^{path'} \\ \vdots \\ d_{N_{path}}^{path'} \end{bmatrix} + \mathbf{W}\mathbf{I}_{wd}, \quad \mathbf{W} = \begin{bmatrix} w_{1,1} \ldots w_{1,n} \\ \vdots \ddots \vdots \\ w_{N_{path},1} \ldots w_{N_{path},n} \end{bmatrix} \tag{4.25}$$

$$w_{j,i} = \begin{cases} K_i^{cell}(t) & \text{if cell } i \text{ is on } p_j \\ 0 & \text{otherwise} \end{cases}$$

[55] The within-die I_{eff} variation can also be estimated from historical data.

where N_{path} is the total number of paths, n is the total number of cell instances and $\mathbf{I}_{wd}$ represents the within-die I_{eff} variation. **W** is a parameter that describes dependencies between critical paths and $\mathbf{I}_{wd}$. Every entry in $\mathbf{I}_{wd}$ is an independent Gaussian random variable, with zero mean and standard deviation σ_{wd}. Due to large numbers of critical paths and cell instances, keeping the entire covariance matrix on test machines is not practical. To reduce the size of **W**, we extract and use its N_{pc} largest principal components (PC). This reduces the total data size by a factor of N_{pc}/N_{path} but some correlation information is lost and the variance of each path delay is less than the exact correlation value. To ensure that we do not underestimate the variance of path delays, difference between **W** and **W'** is represented as a residue term r_j for each path. This residue is assumed to be uncorrelated such that it is unlikely to underestimate the path delay. Therefore, the path delays can be expressed as

$$\mathbf{D} = \begin{bmatrix} d_1^{path'} \\ \vdots \\ d_{N_{path}}^{path'} \end{bmatrix} + \mathbf{W'I}_{wd} + \sum_{j=1}^{N_{path}} r_j \tag{4.26}$$

where **W'** is the compressed matrix with N_{pc} principal components. Though part of the correlation information is not captured, Figure 4.17 shows that our method is efficient in reducing

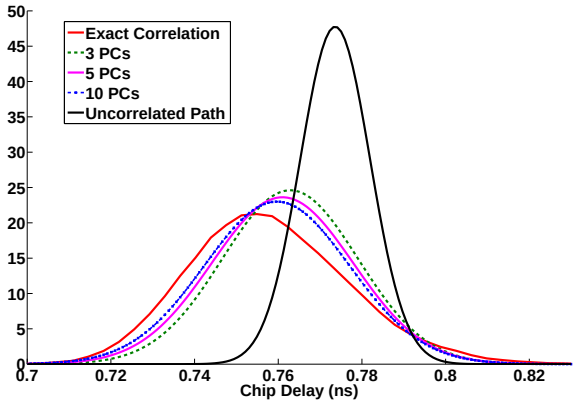


Figure 4.17: Comparison between delay distributions for circuit *C432*.

pessimism in delay estimation, in contrast to assuming that all paths are completely independent. Moreover, this method is flexible as it provides a tradeoff between accuracy and data size, by choosing a suitable number of principal components. The size of correlation matrix is O($N_{pc} \times N_{path}$).

In Equation (4.26), each row of **D** represents delay of a path in the canonical form for tightness probability calculation. We use the method proposed in [202] to obtain the maximum delay of N_{path} critical paths on a chip.

$$d^{chip} = \mathcal{N}(\mu_{\text{delay}}, \sigma_{\text{delay}}) \tag{4.27}$$

where d^{chip} is the maximum delay of a chip, and μ_{delay} and σ_{delay} are the mean and standard deviation of maximum delay distribution of a chip.

Dealing with Measurement Noise

To reduce the measurement uncertainties, it is common to have multiple devices under test connected in parallel and carry out the measurement repeatedly. Thus, we assume every measurement is repeated N_e times, and the scribeline test structure has N_d devices connected in parallel. Only the sum of device currents and capacitance of every chip are measured, i.e., the mean I_{eff}, I_{off} and device capacitance per unit width are obtained. The mean of measured I_{eff} for a chip is denoted as $\hat{I}_{eff}$, and it is expressed as

$$\hat{I}_{eff} = \frac{1}{N_e} \sum_{m=1}^{N_e} \frac{\tilde{I}_{eff}(m)}{N_d} \tag{4.28}$$

where $\tilde{I}_{eff}(m)$ is the sum of I_{eff} for N_d devices at the m^{th} measurement and N_e is the total number of measurements. Based on the measured $\hat{I}_{eff}$, we can represent I_{eff} as follows (see Appendix A for detailed derivations).

$$\begin{aligned} \mu_{I_{eff}} &= \hat{I}_{eff} \\ \sigma^2_{I_{eff}} &= \frac{\hat{I}_{eff}\sigma^2_{I_{wd}}}{N_d} + \frac{\sigma^2_{Z\_eff}}{N_e} \end{aligned} \tag{4.29}$$

where $\sigma^2_{I_{wd}}$ and $\sigma^2_{Z\_eff}$ are the variance of within-die variation and measurement noise, respectively. Note that the variance of I_{eff} is inversely proportional to the number of measurements and total devices in the test structure. Unless otherwise mentioned, we assume five measurements are taken every time ($N_e = 5$) and there are 10 devices in each test structure ($N_d = 10$). We assume that $3 \times \sigma_{Z\_eff}$ is 5% of nominal I_{eff} value. $\sigma_{I_{wd}}$ is obtained by running Monte Carlo simulation over the variation ranges specified in Table 4.7.

Interconnect Delay Variation

Since scribeline measurement is done after M1 layer, the proposed model cannot fully capture interconnect-induced delay variation. However, the effect of interconnect variation is less pronounced due to the following reasons [32].

- Interconnect variations on different metal layers are independent. Therefore, interconnect-induced delay variation averages out to a small value when a path passes through different metal layers.
- Interconnect width variation changes wire resistance and capacitance in opposite ways, thus reducing its net effect on RC.

Nonetheless, we include the effect of interconnect variation in our experiments and measure the error incurred in estimation of delay.

4.2.2 Leakage Power Estimation Using I_{off}

Leakage Power Model

We model leakage power of a chip (P^{chip}) as a linear function of I_{off} as follows.[56]

$$P^{chip} = \sum_{t \in T} \sum_{c \in \Gamma} \sum_{l \in G_c} \alpha(c,t) I_{off}(l,c,t) \tag{4.30}$$

where l is the index for an instance, T is the set of device types, G_c is the set of instances for cell type c in the design, and Γ is the set of all cell types. $\alpha(c,t)$ is the leakage power fitting coefficient for cell type (c) and device type (t). $I_{off}(l,c,t)$ is leakage current of for an instance l with cell type c and device type t. To estimate leakage power variation, we model I_{off} as an exponential function of variation sources [173].

$$I_{off}(l,c,t) = I_{off\_nom}(c,t) e^{Y(l,c,t)}$$

where I_{off_nom} is the nominal I_{off} and Y represents the impact of variation sources. We model Y as a linear combination of inter-die and within-die variations, which are Gaussian random variables,

$$I_{off}(l,c,t) = I_{off\_nom}(c,t) e^{Y_g(t) + Y_r(l,c,t)} \tag{4.31}$$

[56] We only consider subthreshold leakage, but the model can be easily extended to consider gate leakage.

where $Y_g(t)$ denotes the total inter-die variation for device type t. $Y_r(l,c,t)$ is the within-die variation for device type t in cell type c and is specific to instance l. Combining Equations (4.30) and (4.31), we have

$$\begin{aligned} P^{chip} &= \sum_{t\in T}\sum_{c\in\Gamma} P^{cell}(c,t) \\ P^{cell}(c,t) &= \alpha(c,t) I_{off\_nom}(c,t) e^{Y_g(t)} \sum_{l\in G_c} e^{Y_r(l,c,t)} \\ &\text{since } \sum_{l\in G_c} e^{Y_r(l,c,t)} \approx |G_c|\cdot\mu_r(c,t) \text{ [173]} \\ P^{cell}(c,t) &\approx \alpha(c,t) I_{off\_nom}(c,t) e^{Y_g(t)} |G_c|\cdot\mu_r(c,t) \end{aligned} \tag{4.32}$$

where P^{cell} is total leakage power of cell type c for a chip, $|G_c|$ is the total number of instance of cell type c in the chip, $\mu_r(c,t)$ is the mean of $e^{Y_r(l,c,t)}$, which the foundry can extract from historical data. In our experiments, $\mu_r(c,t)$ is obtained by running Monte Carlo simulations at randomly sampled process conditions, based on the variation model in Table 4.7 (see Section 4.2.4).

Dealing with Measurement Noise

To calculate leakage power of a die, we extract $Y_g(t)$ by measuring $I_{off}(t)$ of N_d devices of type t for N_e times. I_{off} of the m^{th} measurement of device type t is modeled as follows.

$$\begin{aligned} \tilde{I}_{off}(m,t) &= \sum_{s=1}^{N_d} I_{off\_nom}(t) e^{Y_g(t)+Y_{rt}(s,t)}(1+Z_{off\_m}) \\ &\approx N_d I_{off\_nom}(t) e^{Y_g(t)} \mu_{rt}(1+Z_{off\_m}) \end{aligned} \tag{4.33}$$

where $\tilde{I}_{off}(m,t)$ is the sum of I_{off} for N_d devices of type t in the m^{th} measurement, and $\mu_{rt}(t)$ is the mean of $e^{Y_{rt}(s,t)}$. Z_{off_m} is the normalized measurement noise in the m^{th} measurement, which is modeled as a Gaussian random variable with zero mean and standard deviation σ_Z. Based on the measured leakage current, the mean ($\mu_{Y_g(t)}$) and variance ($\sigma^2_{Y_g(t)}$) of $Y_g(t)$ are given as follows (see Appendix B for details).

$$\begin{aligned} \mu_{Y_g(t)} &= \frac{1}{N_e}\sum_{m=1}^{N_e} ln\left(\frac{\tilde{I}_{off}(m,t)}{N_d I_{off\_nom}(t)\mu_{rt}}\right) \\ \sigma^2_{Y_g(t)} &= \sigma_Z^2/N_e \end{aligned} \tag{4.34}$$

Equation (4.32) shows that P^{chip} is the sum of $P^{cell}(c,t)$, each of which is a lognormal distribution.[57] Thus, we can apply Wilkinson's approach [173] to approximate P^{chip} as a lognormal random variable, and calculate its mean and variance based on the lognormal distribution of P^{cell} specified by $Y_g(t)$.

4.2.3 Wafer and Chip Pruning Strategy

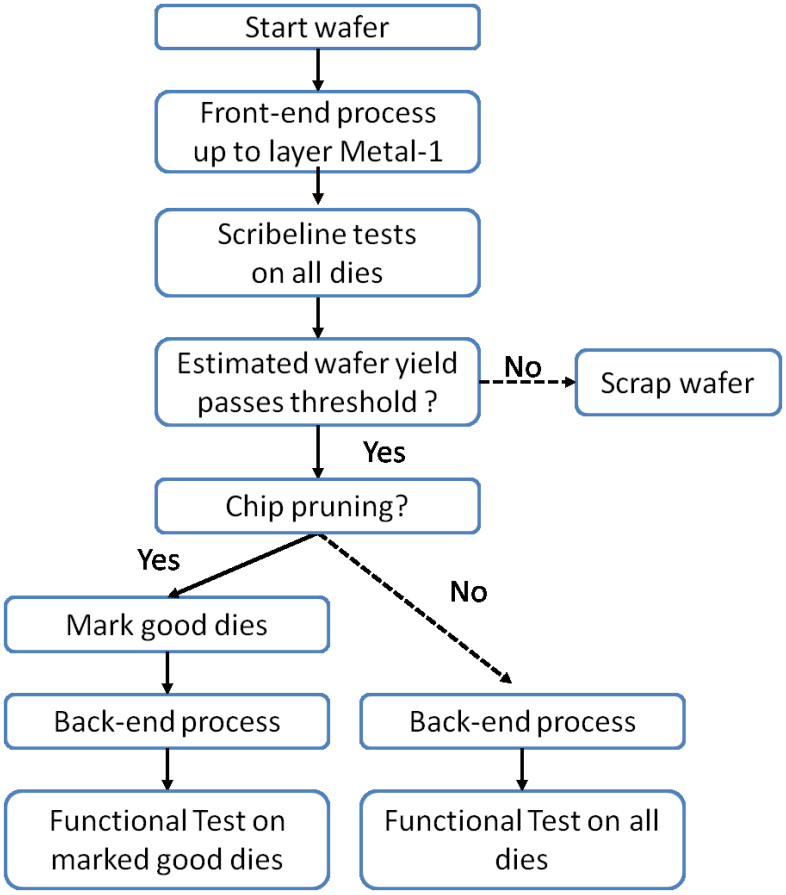


Figure 4.18: Proposed wafer and chip pruning flow.

In conventional manufacturing, accurate circuit performance becomes available only after dicing and packaging. Any failed chip at that stage incurs losses due to unnecessary fabrication, packaging, and testing costs. To reduce the cost per good chip, we propose a wafer and chip pruning flow illustrated in Figure 4.18. After processing a wafer up to layer M1, scribeline

[57] $Y_g(t)$ for all device types is affected by within-die random variation and measurement noise, which are mutually independent. Therefore, the mean and variance of P^{chip} can be calculated as the sum of the mean and variance of $P^{cell}(c,t)$.

measurements are carried out on every die. Based on the scribeline measurement data, we estimate chip performance and calculate the expected yield of each wafer. A wafer will be scrapped if the expected number of good chips does not meet a predefined wafer pruning threshold (WPT) value. For the *wafer-level pruning only* scenario, wafers that pass the pruning threshold will go through back-end process and functional test, as in conventional manufacturing flow. For *wafer and chip pruning* scenario, good dies are marked using existing techniques (e.g., [12] [55]) such that only the good dies will be tested after back-end processes.

Passing Probability for a Chip

Given the measured I_{eff} and capacitance, conditional probability of a chip meeting timing constraint is given by

$$\begin{aligned} &\mathbb{P}\left\{d^{chip} \leq d_{spec} | (I_{eff} = \hat{I}_{eff}, C_{gate} = \hat{C}_{gate})\right\} \\ &\quad = \Phi(\frac{d_{spec} - \mu_{\text{delay}}}{\sigma_{\text{delay}}}) \end{aligned} \tag{4.35}$$

where $\hat{C}_{gate}$ is the measured capacitance, $\hat{I}_{eff}$ is the mean of measured I_{eff}, d_{spec} is the maximum allowed delay for a design, $\Phi(\cdot)$ is the standard normal cumulative distribution function, μ_{delay} and σ_{delay} are mean and standard deviation of maximum chip delay distribution. On the other hand, the probability of a chip meeting leakage power constraint is given by

$$\begin{aligned} &\mathbb{P}\left\{P^{chip} \leq P_{spec} | I_{off} = \hat{I}_{off}\right\} \\ &\quad = \mathbb{P}\left\{ln(P^{chip}) \leq ln(P_{spec}) | I_{off} = \hat{I}_{off}\right\} \\ &\quad = \Phi[\frac{ln(P_{spec}) - \mu_L}{\sigma_L}] \end{aligned} \tag{4.36}$$

where μ_L is the mean of $ln(P^{chip})$ and σ_L is the variance of $ln(P^{chip})$. P_{spec} is the maximum allowed leakage power for a design.

Given the measured values ($\hat{I}_{eff}$, $\hat{I}_{off}$ and $\hat{C}_{gate}$) of every chip, the probability of a chip meeting timing or leakage power constraint is determined by the uncertainties in chip delay and leakage power. Note that uncertainty in delay estimation (σ_{delay}) is due to I_{eff} within-die variation and measurement noise, while uncertainty in leakage power estimation (σ_L) is only induced by measurement noise in I_{off}. Since the measurements of I_{eff} and I_{off} are taken using different measurement steps and bias conditions, the measurement noise for leakage power estimation is independent of the measurement noise for delay estimation. As a result, the

uncertainties of chip delay and leakage power are modeled by two independent Gaussian random variables. Therefore, the probability of a chip meeting the timing constraint and the probability of a chip meeting the leakage power constraint are **conditionally independent** given the values of $\hat{I}_{eff}$, $\hat{I}_{off}$ and $\hat{C}_{gate}$. The passing probability of a chip is given by

$$\begin{aligned}
&\mathbb{P}\left\{P^{chip} = \text{pass} | (I_{eff} = \hat{I}_{eff}, C_{gate} = \hat{C}_{gate}, I_{eff} = \hat{I}_{eff})\right\} \\
&= \mathbb{P}\left\{P^{chip} \leq P_{spec} | I_{off} = \hat{I}_{off}\right\} \times \\
&\quad \mathbb{P}\left\{d^{chip} \leq d_{spec} | (I_{eff} = \hat{I}_{eff}, C_{gate} = \hat{C}_{gate})\right\}
\end{aligned} \tag{4.37}$$

Meanwhile, the expected number of good chips in a wafer ($N_{c\_good\_est}$) can be estimated as the sum of passing probability of all chips in a wafer.

$$\begin{aligned}
&N_{c_good_est} = \\
&\sum_{\text{chips}} \mathbb{P}\left\{P^{chip} = \text{pass} | (I_{eff} = \hat{I}_{eff}, C_{gate} = \hat{C}_{gate}, I_{eff} = \hat{I}_{eff})\right\}
\end{aligned} \tag{4.38}$$

Cost Model

The benefit of wafer or chip pruning is related to chip selling price, manufacturing cost and testing cost, which are affected by many factors. For example, the chip selling price varies due to demand and supply of a product, marketing strategy, etc.; manufacturing cost depends on manufacturing equipment, raw materials, and processing costs [229]; testing cost is affected by the number of test patterns and the testing infrastructure. Table 4.6 shows the relative costs for scribeline testing (M_s), *front-end-of-line* (M_f), back-end-of-line (M_b), and full-chip testing cost (M_t) for different scenarios. For cost setup 1, we obtain the ratio between M_f and M_b from [207]. The cost model in [207] describes a wafer process with 20 layers, and processing each layer costs \$466. We assume that the front-end cost, M_f, includes the processing cost for the first 10 layers of a wafer, and a \$81.6/wafer raw wafer cost [207]; M_b includes the processing cost for the remaining 10 layers. We then estimate the testing cost, M_t, as 50% of the total manufacturing cost (M_f + M_b) [172]. Cost setups 2 and 3 are hypothetical cases to evaluate the benefit of proposed wafer pruning for different cost setups.

We assume that the scribeline testing cost is negligible in cost setups 1, 2, and 3, as scribeline measurements may be taken by a foundry as a standard procedure for process monitoring. Cost setups 4, 5 and 6 model the scenario where scribeline measurements are not taken in the standard manufacturing flow and the measurements incur additional cost. We assume that

Table 4.6: Manufacturing and testing cost setups, where the costs are represented in percentages.

	Setup 1	Setup 2	Setup 3	Setup 4	Setup 5	Setup 6
Scribeline test cost (%)	0	0	0	3	3	3
Front-end cost (%)	36	60	20	35	59	19
Back-end cost (%)	30	20	20	29	19	19
Test cost (%)	34	20	60	33	19	59
Total cost (%)	100	100	100	100	100	100

scribeline measurement cost is lower than the final testing cost because the number of items to be measured is much less than the final testing ones.

We acknowledge that our cost model does not consider many practical aspects of semiconductor manufacturing. However, the cost model mainly affects wafer pruning threshold (WPT), which is determined by fixed cost (irrespective of pruning) and pruning-dependent cost. Therefore, we split the total manufacturing cost into four components that are fixed or pruning-dependent, and evaluate several scenarios by varying the relative values among the cost components. The actual pruning decision making and WPT will depend on variety of factors, including cost, volume demand, machine capacity, chip price, etc., detailed analysis of which are beyond the scope of this thesis.

Wafer and Chip Pruning Analysis

In the proposed wafer pruning strategy, we will prune a wafer if its expected yield is lower than WPT. Clearly, the benefit of pruning is dependent on the WPT value, which can be guided by the expected profit and additional cost to continue making the wafer. We define WPT such that we will prune a wafer only if its expected profit is smaller than additional cost to make the wafer. The WPT for two pruning scenarios are given as follows.

- Option 1: wafer pruning only

$$\begin{aligned} &\text{Additional Cost} = (M_b + M_t) \\ &\text{Expected profit} = N_{c\_good\_est} \times \text{Chip price} \\ &\text{Expected profit} > \text{Additional Cost} \\ &N_{c\_good\_est} \times \text{Chip price} > (M_b + M_t) \\ &\implies \text{WPT} = \frac{(M_b + M_t)}{\text{Chip price}} \end{aligned} \tag{4.39}$$

- Option 2: wafer and chip pruning

$$\begin{aligned}
&\text{Additional Cost} = (M_b + N_{c_good_est} \times M_t) \\
&\text{Expected profit} = N_{c_good_est} \times \text{Chip price} \\
&\text{Expected profit} > \text{Additional Cost} \\
&N_{c_good_est} \times (\text{Chip price} - M_t) > M_b \\
&\implies \text{WPT} = \frac{(M_b)}{\text{Chip price} - M_t}
\end{aligned} \tag{4.40}$$

Note that we do not consider the cost for front-end processes in Equations (4.39) and (4.40) because the process has been carried out and incurred processing cost regardless of the pruning decision. The chip selling price is also a factor during wafer pruning. For example, if the chip selling price is much larger than the total manufacturing cost, then the foundry is less likely to prune a wafer because its expected profit is always greater than the additional cost to make a wafer. When we combine wafer and chip pruning, the additional cost to manufacture a wafer is lower because only a subset of the chips will be tested. Thus, WPT for combined wafer and chip pruning is less than the WPT of the wafer pruning only scenario.

4.2.4 Experimental Results

Figure 4.19 summarizes our experiment setup, which demonstrates the flow of the proposed wafer pruning method. The upper part of the figure describes procedures to obtain design-specific parameters at a design house. We use Monte Carlo SPICE simulations with the variation model specified in Table 4.7 to generate samples for $K_i^{cell}(c,t)$ and $\alpha(c,t)$ characterization. Note that the Monte Carlo SPICE simulation can be replaced by timing libraries at various process corners to speed up the characterization. We characterize $K_i^{cell}(c,t)$ and $\alpha(c,t)$ with the $45nm$ *Nangate Open Cell library* [240].

We implement a combination of *ISCAS85* and *OpenCores* benchmark circuits with the $45nm$ *Nangate Open Cell library*. We extract the critical paths of the benchmark circuits and G_c. We consider all paths with nominal delay within 5% of the maximum path delay as critical paths.[58] Based on the nominal slew and load on critical paths, we compute $K_j^{path}(t)$, **W'**, **R**, $|G_c|$ and $\sum_c \sum_t \{\alpha(c,t)\}$ coefficients. These compressed design-dependent coefficients will be used to estimate chip delay and leakage power for the proposed pruning strategy.

[58] Many improved critical path selection algorithms have been proposed in literature [210] [222]. We do not implement the path selection algorithms, as it is beyond the scope of this thesis.

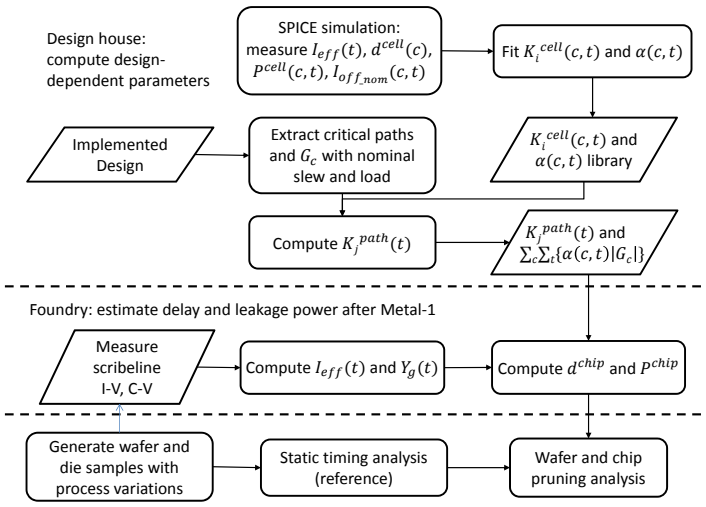


Figure 4.19: Proposed delay and leakage power estimation method.

Due to the lack of foundry data, we simulate wafer and die samples based on the variation model in Table 4.7. For every benchmark circuit, we simulate 250 wafers, each of which has 657 chips. For each chip, we obtain its delay and leakage power by using *Synopsys PrimeTime* [254]. If the delay is smaller than d_{spec} and the leakage power is smaller than P_{spec}, the chip is considered to be a good chip.

At the same time, we simulate PMOS and NMOS devices (high V_{th} and low V_{th}) using SPICE to extract I_{eff} and I_{off} (to emulate scribeline measurements). The devices have the same inter-die variation values as the chip, but there are mismatches due to within-die variation. For I_{eff} extraction, we use five principal components for each device type. Based on the simulated I_{eff} and I_{off} we compute the d^{chip} and P^{chip} of every chip. We perform STA and power analysis on the chip samples to obtain actual delay and leakage power for the wafer pruning benefit calculation. To evaluate the benefit of design-dependent delay and leakage power models, we implement a design-independent approach, which equally weighs high V_{th} and low V_{th} devices in the delay and leakage power estimations. We assume that the design-independent delay estimation is inversely proportional to the mean of I_{eff} of all device types. Similarly, the leakage power estimation is proportional to the mean of I_{off} of all device types. Unless otherwise specified, timing constraints of the benchmark circuits are 110% of the nominal critical path delay of the respective designs, and the leakage power constraints are five times the nominal leakage power.

Table 4.7: Summary of variation parameters.

Variation source	Wafer-to-wafer (%)	Die-to-die (%)	Die-to-die (%)	Within-die (%)
Variation type	Random	Systematic	Random	Random
Channel length	$\mathcal{N}(0, 2.13)$	$q_1 \cdot x^2 + q_2 \cdot y^2 + q_3 \cdot x + q_4 \cdot y + q_5 \cdot x \cdot y$	$\mathcal{N}(0, 1.29)$	$\mathcal{N}(0, 1.56)$
NMOS V_{th}	$\mathcal{N}(0, 6.4)$	–	$\mathcal{N}(0, 6.08)$	$\mathcal{N}(0, 4.7)$
PMOS V_{th}	$\mathcal{N}(0, 6.4)$	–	$\mathcal{N}(0, 6.08)$	$\mathcal{N}(0, 4.7)$
Interconnect width	–	–	$\mathcal{N}(0, 10)$	–
Interconnect thickness	–	–	$\mathcal{N}(0, 10)$	–

Variation Model

We model five independent variation sources for transistors as shown in Table 4.7. V_{th} variations are modeled as Gaussian distributed random variables with no spatial variation [220]. Channel length is assumed to be the only variation source, which contributes to systematic delay variation across wafer. The across-wafer systematic delay variation Δd_{sys} is modeled as

$$\Delta d_{sys} = q_1 \cdot x^2 + q_2 \cdot y^2 + q_3 \cdot x + q_4 \cdot y + q_5 \cdot x \cdot y, \tag{4.41}$$

where x and y represent the coordinates of a chip's centroid [49]. The wafer diameter is $300mm$ and 657 chip centroids are distributed uniformly across the wafer. Since the model is applicable from $90nm$ to $45nm$ technologies [49] [171], we obtain the values of q_1, q_2, q_3, q_4 and q_5 by matching across-wafer systematic delay variation to $65nm$ silicon data.[59] V_{th} variations in Table 4.7 are also extracted from the same silicon data. To model interconnect variation, we obtain σ/μ ratio of wire width from [231], and assume that wire thickness has a similar ratio.[60]

Interconnect variation is modeled as random Gaussian-distributed intra-die variation [29]. In our experiments, this is implemented by perturbing unit resistance and capacitance values in the LEF files of implemented benchmark circuits.

[59] For our model, $q_1 = 7.7e^{-4}$, $q_2 = 1.0e^{-3}$, $q_3 = -1.6e^{-2}$, $q_4 = -7.8e^{-3}$, $q_5 = 1.6e^{-4}$.

[60] Wire thickness variation is not available in ITRS reports.

Table 4.8: Cost comparison for chip selling price = 1.5 times of the cost per chip with 100% yield (normalized to the cost per chip with 100% yield). *Dep.*, *Indep.* and *Nom.* refer to design-dependent, design-independent and no pruning experiment setups, respectively.

Benchmarks	Cost setup 1			Cost setup 2			Cost setup 3			Cost setup 4			Cost setup 5			Cost setup 6		
	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.
C432	1.54	1.59	1.62	1.67	1.66	1.62	1.45	1.54	1.62	1.55	1.59	1.62	1.68	1.67	1.62	1.47	1.55	1.62
C432L	1.26	1.34	1.29	1.29	1.41	1.29	1.24	1.28	1.29	1.26	1.34	1.29	1.29	1.42	1.29	1.24	1.29	1.29
S15850	1.40	1.44	1.48	1.50	1.51	1.48	1.33	1.39	1.48	1.41	1.45	1.48	1.51	1.52	1.48	1.34	1.40	1.48
S38584	1.33	1.39	1.36	1.42	1.46	1.36	1.27	1.34	1.36	1.33	1.39	1.36	1.42	1.47	1.36	1.27	1.34	1.36
MIPS	1.34	1.42	1.37	1.41	1.48	1.37	1.29	1.38	1.37	1.34	1.42	1.37	1.42	1.48	1.37	1.29	1.38	1.37
Average	1.37	1.43	1.43	1.46	1.50	1.43	1.32	1.39	1.43	1.38	1.44	1.43	1.46	1.51	1.43	1.32	1.39	1.43

Table 4.9: Cost comparison for chip selling price = 1.7 times of the cost per chip with 100% yield (normalized to the cost per chip with 100% yield). *Dep.*, *Indep.* and *Normal* refer to design-dependent, design-independent and no pruning experiment setups, respectively.

Benchmarks	Cost setup 1			Cost setup 2			Cost setup 3			Cost setup 4			Cost setup 5			Cost setup 6		
	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.	Dep.	Indep.	Nom.
C432	1.53	1.58	1.62	1.64	1.64	1.62	1.47	1.55	1.62	1.54	1.59	1.62	1.64	1.64	1.62	1.47	1.55	1.62
C432L	1.26	1.32	1.29	1.28	1.38	1.29	1.24	1.28	1.29	1.26	1.33	1.29	1.29	1.39	1.29	1.25	1.29	1.29
S15850	1.40	1.44	1.48	1.48	1.50	1.48	1.35	1.40	1.48	1.41	1.45	1.48	1.49	1.50	1.48	1.35	1.41	1.48
S38584	1.32	1.38	1.36	1.39	1.44	1.36	1.27	1.34	1.36	1.33	1.38	1.36	1.40	1.44	1.36	1.28	1.34	1.36
MIPS	1.33	1.40	1.37	1.39	1.45	1.37	1.29	1.37	1.37	1.34	1.41	1.37	1.39	1.45	1.37	1.30	1.38	1.37
Average	1.37	1.43	1.43	1.44	1.48	1.43	1.32	1.39	1.43	1.37	1.43	1.43	1.44	1.49	1.43	1.33	1.39	1.43

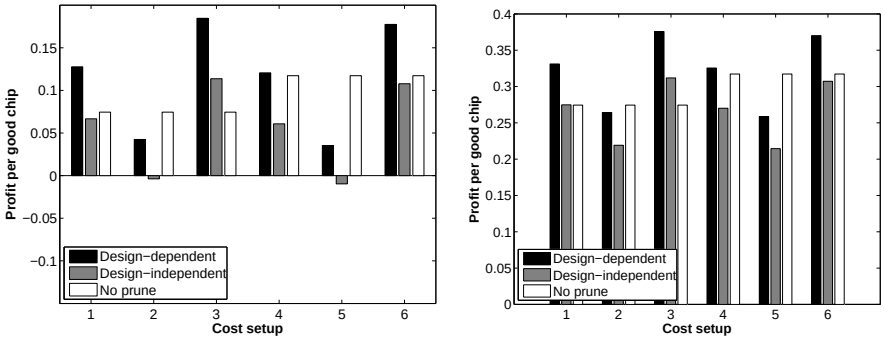
Wafer Pruning Results

In Table 4.8 and Table 4.9, we compare the *cost per good chip* resulting from the proposed wafer pruning method. The cost per good chip are defined as follows.

$$\text{cost per good chip with no pruning} = \frac{(M_f + M_b + M_t) \times N_w}{N_{c\_good\_act}}$$
$$\text{cost per good chip with pruning} = \frac{(M_s + M_f) \times N_w + (M_b + M_t) \times N_{w\_good}}{N_{c\_good\_act}} \tag{4.42}$$

where N_w is the total number of wafer ($N_w = 250$) and $N_{c\_good\_act}$ is the total number of actual good chips. N_{w_good} is the total number of wafers with a yield rate (ratio of $N_{c\_good\_act}$ to total chips on a wafer) higher than the WPT. $N_{c\_good\_act}$ is obtained by summing up actual good chips for wafers that pass the early wafer pruning. Note that N_{w_good} varies depending on the pruning method. Therefore, the $N_{c\_good\_act}$ is also different across the pruning methods.

Table 4.8 and Table 4.9 show that the cost per good chip is higher than 1.0 for no wafer pruning case. This happens because the wafer yield is smaller than 100% (due to process variation). Results in the tables show that proposed design-dependent wafer pruning method reduces cost per good chip by up to 10% compared to the no pruning case when a large portion of the total cost is spent on back-end and final testing (cost setups 1, 3, 4, and 6). When wafer cost is dominated by front-end and fixed costs (cost setups 2 and 5), wafer pruning may increase the total cost. On an average, design-dependent wafer pruning can reduce cost per good chip by 6%, compared to the design-independent wafer pruning approach.



(a) Normalized chip selling price = 1.5 (b) Normalized chip selling price = 1.7

Figure 4.20: Average profit per good chip of all benchmarks with different cost setups. Profit per good chip and chip selling price are normalized to the cost per chip with 100% yield.

Table 4.10: Cost per good chip (normalized to the cost per chip with 100% yield) for design-dependent wafer pruning based on limited sampling. Chip selling price is 1.7 times the cost per chip with 100% yield.

Sampling ratio (%)	5	10	30	50	80	100
Cost setup 1	1.36	1.37	1.38	1.38	1.38	1.38
Cost setup 2	1.54	1.54	1.54	1.54	1.54	1.54
Cost setup 3	1.19	1.18	1.18	1.17	1.18	1.18
Cost setup 4	1.34	1.34	1.36	1.37	1.38	1.39
Cost setup 5	1.50	1.51	1.52	1.53	1.54	1.55
Cost setup 6	1.16	1.16	1.17	1.18	1.19	1.20

Figure 4.20 shows the profit per good chip for different pruning approaches and cost setups.

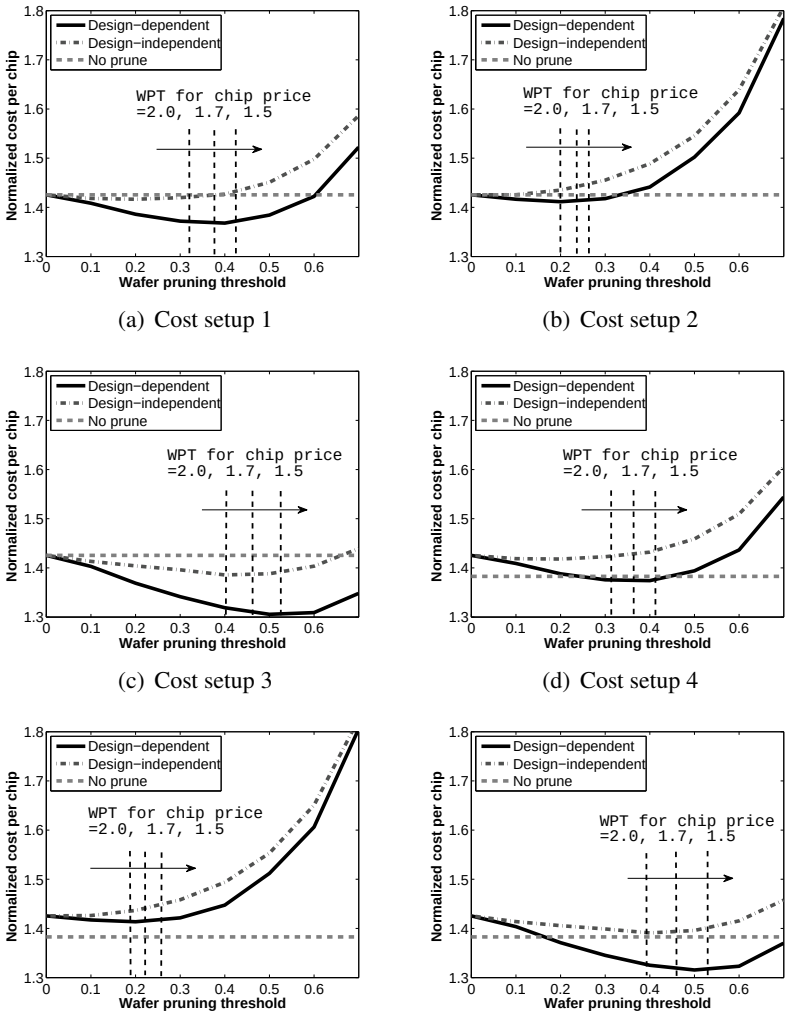
$$\text{Profit per good chip} = \quad \text{chip selling price} - \text{cost per good chip} \tag{4.43}$$

The results show that the proposed design-dependent method has a higher profit per good chip compared to the design-independent method. Early wafer pruning is beneficial when wafer cost is dominated by back-end processes and final test cost (cost setups 1, 3, 4 and 6). However, early wafer pruning reduces profit per good chip compared to the no pruning case, when wafer cost is dominated by front-end and fixed costs (cost setups 2 and 5).

Figure 4.21 shows optimal WPT that minimizes cost per good chip varies for different cost setups. Therefore, we need to set WPT according to the cost setups. When wafer cost is dominated by back-end and test costs (cost setups 3 and 6), we need to set a larger WPT such that any manufactured wafer has enough good chips to compensate for manufacturing and test cost. When wafer cost is dominated by front-end and fixed costs (cost setups 2 and 4), we need to set a lower WPT because scrapping any wafer incurs significant losses. As chip selling price reduces, the expected profit of making a wafer also reduces. As a result, a higher WPT is needed to ensure that it is beneficial to continue processing a wafer.

Results in Figure 4.21 also show that the WPT estimated by (4.39) is a good approximation to the optimal WPT that minimizes cost per good chip. When the WPT is large than 0.5, most of the wafers will be pruned even if there are many good chips on a wafer. As a result, the cost per good chip increases along with WPT.

To reduce the scribeline testing cost, we study the impact of randomly sampling chips for delay and leakage power estimation (instead of measuring every chip on a wafer) on wafer pruning quality. In this experiment, we estimate the delay and leakage power based on the



(a) Cost setup 1

(b) Cost setup 2

(c) Cost setup 3

(d) Cost setup 4

(e) Cost setup 5

(f) Cost setup 6

Figure 4.21: Average cost per good chip of all benchmarks with different wafer-level pruning strategies. Cost per good chip and chip selling price are normalized to the cost per chip with 100% yield.

Table 4.11: Cost per good chip (normalized to the cost per chip with 100% yield) of benchmark *C432* for different measurement/test structure setups. Chip selling price = 1.7 times the cost per chip with 100% yield.

N_e	N_d	Cost setup 1	Cost setup 2	Cost setup 3	Cost setup 4	Cost setup 5	Cost setup 6
1	1	1.55	1.69	1.45	1.56	1.70	1.46
5	10	1.54	1.67	1.45	1.55	1.68	1.47
100	100	1.54	1.67	1.45	1.55	1.68	1.47

randomly sampled chips and the scribeline test cost is scaled proportionally with the sampling ratio. Table 4.10 shows that total cost per good chip reduces as the number of samples reduces for cost setups 3, 4, and 5. This implies that the proposed method can minimize cost overhead incurred by scribeline testing. Note that the proposed method can be further improved by other sampling strategies [132] [175].

To evaluate the impact of measurement noise and test structure design, we run an experiment with different N_e and N_d. Table 4.11 shows that the cost per good chip achieved by our strategy is insensitive to the measurement count and to the number of devices in test structures. Therefore, the test structures can be further optimized to reduce measurement time and scribeline area.

Chip Pruning Results

Figure 4.22 shows the chip pruning benefits of the proposed strategy for the *C432* and *MIPS* benchmarks as described in Section 4.2.3. The y-axis shows the percentage of chips that are bad and pruned. The x-axis shows the amount of yield loss that results from chip pruning. The plot is made by varying delay and leakage power guardbands, i.e., we scale the chip's delay by ζ_d and the chip's leakage power by ζ_p. For some points on the plot, we indicate the scaling factors in the parentheses, i.e., (ζ_d, ζ_p). The values in each pair of square brackets are the prune percentage and the yield loss. Figure 4.22 shows that there is a tradeoff between the percentage of chips pruned and yield loss. Note that a very large percentage of bad chips can be efficiently pruned at the cost of very small yield loss, which results in significant savings on the costly tester time. For example, we can prune almost 70% of bad chips with less than 1% yield loss. This corresponds to almost 15% savings on the tester time. Effective chip pruning is only possible if false positive cases (pruned chips are good chips) are less likely to happen compared to true positive cases (pruned chips are bad chips). This happens when the probability of estimation

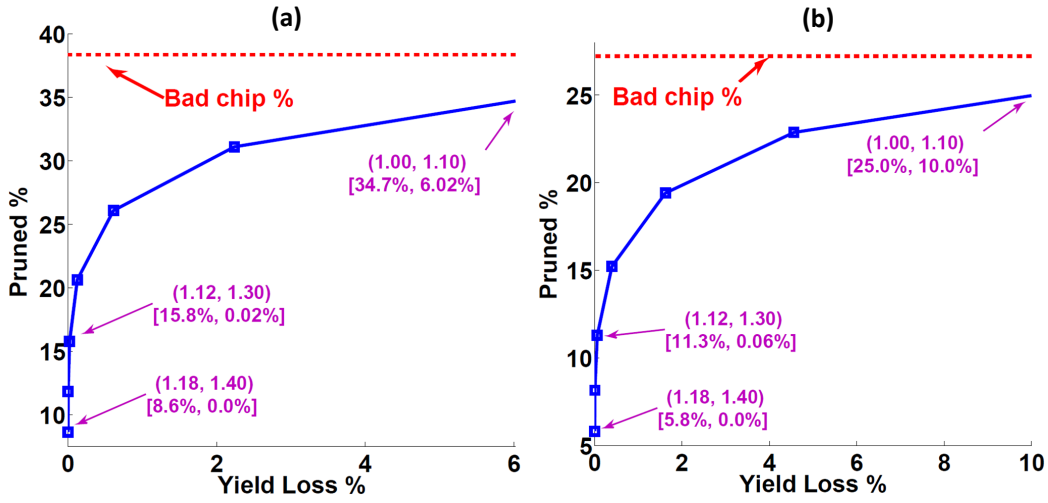


Figure 4.22: Chip pruning results for benchmark design (a) *C432*, (b) *MIPS*. Values in the parentheses are delay and leakage power guardbands. Values in the square brackets are the prune percentage and the yield loss.

error reduces sharply as the magnitude of estimation error increases (e.g., a normal distribution). Table 4.12 shows that experiments on other benchmark circuits show similar chip pruning results.

Figure 4.23 shows that chip pruning can achieve about 5% cost reduction compared to the design-independent approach. Meanwhile, the cost reduction compared to the no-pruning case varies from -1% to 10%, depending on the cost setup. The higher cost of design-independent chip pruning implies that inaccurate performance estimation in the design-independent approach can cause losses when it prunes a good working chip.

Table 4.12: Prune percentage and yield loss of benchmark circuits. The last column indicates total bad chips (%) in all wafers (without wafer pruning).

Guardband	(ζ_d, ζ_p) (1.06, 1.20)		(ζ_d, ζ_p) (1.12, 1.30)		(ζ_d, ζ_p) (1.18, 1.40)		Bad
	Prune %	YL %	Prune %	YL %	Prune %	YL %	chip %
C432	26.08	0.61	15.78	0.02	8.64	0.00	38.40
S15850	23.17	1.27	15.86	0.12	9.94	0.01	32.67
S38584	19.28	2.40	12.12	0.31	6.90	0.03	26.54
MIPS	19.41	1.62	11.29	0.06	5.82	0.00	27.21
C432L	11.71	0.22	5.72	0.02	3.01	0.01	22.24

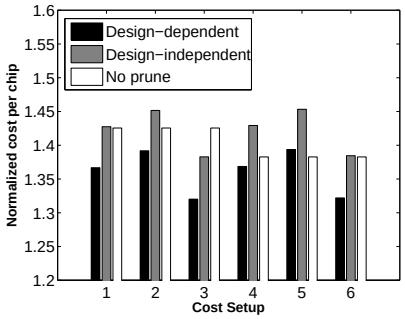


Figure 4.23: Cost per good chip of the average of all benchmark designs using different chip-level pruning strategies. The timing and leakage power guardbands used for chip pruning are 12% and 30%, respectively. Chip selling price is 1.7 times of the cost per chip with 100% yield (WPT = 0).

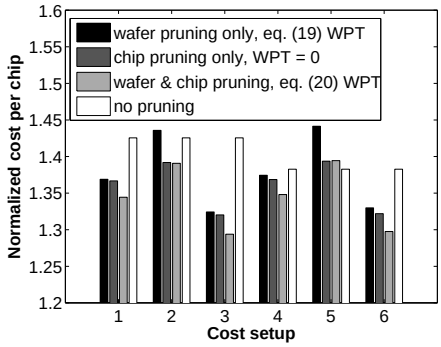


Figure 4.24: Cost per good chip of the average of all benchmark designs using different design-dependent pruning approaches. The chip pruning timing and leakage power guardbands are 12% and 30% of the design's specifications, respectively. Chip selling price is 1.7 times of the cost per chip with 100% yield.

Wafer and Chip Pruning Results

For combined wafer and chip pruning, the cost per good chip is given as follows.

$$\begin{aligned} &\text{cost per good chip} = \\ &\{N_w \times (M_s + M_f) + N_{w\_good} \times M_b \\ &+ N_{c\_good} \times M_t\}/N_{c\_good\_act} \end{aligned} \tag{4.44}$$

where N_{c_good} is the total number of estimated good chips on the wafers that pass WPT. Figure 4.24 shows that when we combine "wafer and chip pruning", the cost per good chip is lower than the no pruning scenario, except in cost setup 5, where most of the cost happens at the early manufacturing stage. In all cases, applying chip-level-only pruning can further reduce cost per

good chip by 1% to 3% compared to applying wafer-level-only pruning because it has a finer pruning granularity. In cases where back-end manufacturing cost dominates the total manufacturing cost (cost setups 3 and 6), wafer-level-only pruning is very effective as it has a similar cost per good chip as that of the wafer and chip pruning method.

4.2.5 Conclusions

We present a novel approach for design-dependent process monitoring. Since the process monitors are placed on wafer scribelines and they can be tested after M1 fabrication, we can perform early chip-performance and wafer-yield estimations for the current process condition (as opposed to long-term process condition statistics). We use the estimations for cutting short the production of obviously bad wafers (i.e., where the wafer yield is too low to cover manufacturing/test costs) and avoiding testing of obviously bad chips. The wafer-pruning approach based on our method can reduce cost per good chip by up to 10%. Using our method, chip pruning can prune almost 70% of bad chips with less than 1% yield loss. Combining the wafer- and chip-pruning methods, we reduce the cost per good chip by 1% to 3% (compared to the wafer pruning only).

4.2.6 Appendix A: I_{eff} Within-Die Variation

We assume that every measurement is repeated N_e times and the scribeline test structure has N_d devices connected in parallel. Only the sum of device currents and capacitance of every chip are measured, i.e., the mean I_{eff}, I_{off} and device capacitance per unit width are obtained. The mean of measured I_{eff} for a chip is denoted as $\hat{I}_{eff}$, and it is given as

$$\hat{I}_{eff} = \frac{1}{N_e} \sum_{m=1}^{N_e} \frac{\tilde{I}_{eff}(m)}{N_d} \tag{4.45}$$

where $\tilde{I}_{eff}(m)$ is the sum of I_{eff} for N_d devices at the m^{th} measurement and N_e is the total number of measurements. Considering measurement noise, $\tilde{I}_{eff}(m)$ can be expressed as

$$\tilde{I}_{eff}(m) = (1 + Z_{eff\_m}) \sum_{s=1}^{N_d} [I_{eff} + I_{wd\_s}] \tag{4.46}$$

where I_{eff} is the exact (unknown) value, I_{wd_s} is the effect of within-die variation on the s^{th} device, and Z_{eff_m} is the m^{th} normalized measurement noise.

Combining Equations (4.28) and (4.46),

$$I_{eff} = \frac{\hat{I}_{eff}}{1+\sum_{m=1}^{N_e} Z_{eff\_m}/N_e} - \frac{1}{N_d}\sum_{s=1}^{N_d} I_{wd_s}$$
$$\because \sum_{m=1}^{N_e} Z_{eff\_m}/N_e \ll 1$$
$$\therefore I_{eff} \approx \hat{I}_{eff}(1+\sum_{m=1}^{N_e} Z_{eff\_m}/N_e) - \frac{1}{N_d}\sum_{s=1}^{N_d} I_{wd_s})$$

Since I_{wd} and Z_eff are Gaussian random variables, I_{eff} is also a Gaussian random variable with its mean and variance given by

$$\mu_{I_{eff}} = \hat{I}_{eff}, \quad \sigma^2_{I_{eff}} = \frac{\hat{I}_{eff}\sigma^2_{I_{wd}}}{N_d} + \frac{\sigma^2_F}{N_e}$$

where $\sigma^2_{I_{wd}}$ and $\sigma^2_{Z\_eff}$ are the variance of the within-die variation and measurement noise for I_{eff}, respectively.

4.2.7 Appendix B: I_{off} Within-Die Variation

Equation (4.32) shows that we need to know Y_g to estimate total leakage power, which is derived from measurements. As mentioned earlier, we take N_e measurements of the current of N_d devices in test structures. Considering measurement noise and within-die variation, the m^{th} measured I_{off} of a given device type t is modeled as

$$\begin{aligned}\tilde{I}_{off}(m,t) &= \sum_{s=1}^{N_d} I_{off\_nom}(t) e^{Y_g(t)+Y_{rt}(s,t)}(1+Z_{off\_m}) \\ &\approx N_d I_{off\_nom}\mu_{rt} e^{Y_g(t)}(1+Z_{off\_m}),\end{aligned} \tag{4.47}$$

where $\tilde{I}_{off}(m,t)$ is the sum of I_{off} for N_d devices at m^{th} measurement, Z_{off_m} is the m^{th} normalized measurement noise. From Equations (4.31) and (4.33), the estimated $Y_g(t)$ is given by

$$\begin{aligned}\hat{Y}_g(t) &= \frac{1}{N_e}\sum_{m=1}^{N_e} ln(\frac{\tilde{I}_{off}(m,t)}{N_d I_{off}(nom)\mu_r t}) \\ &= Y_g(t) + \frac{1}{N_e}\sum_{m=1}^{N_e} ln(1+Z_{off\_m})\end{aligned} \tag{4.48}$$

where $Y_g(t)$ denotes the exact value, $\hat{Y}_g(t)$ is the estimated value. Since the normalized measurement noise Z_{off_m} is much smaller than 1, Equation (4.48) can be simplified as

$$\hat{Y}_g(t) = Y_g(t) + \frac{1}{N_e}\sum_{m=1}^{N_e} Z_{off\_m}, \text{ or}$$
$$Y_g(t) = \hat{Y}_g(t) - \frac{1}{N_e}\sum_{m=1}^{N_e} Z_{off\_m}$$

From the above equation, we observe that the exact inter-die variation $Y_g(t)$ is a random variable centered at $\hat{Y}_g(t)$. Since Z_{off_m} are Gaussian random variables, $Y_g(t)$ is a Gaussian random variable given $\hat{Y}_g(t)$ is a Gaussian random variable. The mean and variance of $Y_g(t)$ are

$$\begin{aligned} \mu_{Yg}(t) &= \hat{Y}_g(t) \\ \sigma^2_{Yg}(t) &= \sigma^2_Z/N_e. \end{aligned} \tag{4.49}$$

Since each $Y_g(t)$ is a Gaussian random variable, $e^{Y_g(t)}$ is a lognormal distribution. From Equation (4.32), we find that P^{chip} is the sum of lognormal distribution. Thus, we can apply Wilkinson's approach [173] to approximate the sum of lognormal random variables as another lognormal random variable by matching the mean and variance.

4.3 BEOL Layout Decomposition with LELE Double Patterning

In litho-etch-litho-etch (LELE) double-patterning lithography (DPL), layout patterns are decomposed into two masks – denoted henceforth as Color 1 and Color 2 – such that all polygons on a given mask satisfy an inter-polygon *minimum coloring spacing* requirement. If a spacing violation, or *coloring conflict*, arises during decomposition, a polygon (net) can be split into two different-color segments to resolve the violation; this introduces a *stitch* where the two segments are overlapped to avoid disconnection due to overlay and/or line-end shortening. Each segment has different parasitic resistance (R) and capacitance (C), and a stitch also affects total RC delay values of the net, depending on its color, geometric dimensions, overlay, stitching location and length, etc.

We study the impact of stitch insertion on interconnect RC as well as on circuit performance. Our motivation is that Color 1 and Color 2 interconnect segments have independent CD distributions (*bimodal CD distribution*) due to two independent exposures in DPL. Gupta et al.

[84] note that bimodality of CD variation on poly-silicon features causes delays across spatially-adjacent transistors have less correlation. When a signal path passes through the transistors, its delay variability is reduced due to the averaging of uncorrelated transistor delays. Following the observation that bimodality of CD variation can reduce delay variability, we study the impact of stitching insertion, which induces bimodality in interconnects.

Conventional layout decomposition algorithms [51] [52] [107] focus on solving the color assignment problem, and ignore the impact of stitches on circuit performance. Yang et al. [213] propose a multi-objective layout decomposition framework that accounts for circuit timing. In their algorithm, stitching locations are defined based on the result of initial layout segmentation. Their experimental results show that introducing more stitches (at arbitrary locations on interconnect) reduces circuit delay variation. However, detailed analysis for stitch insertion is not discussed. Oosten et al. [162] study overlay margin in stitch insertion but do not extend their work on the impact of stitching on circuit performance.

Our studies using $45nm$ (commercial) and $22nm$ (ITRS) technology parameters show that 3σ delay variation varies by as much as 5% when a stitch location is swept along an interconnect. We notice that delay variations are higher when the stitch is located at the driver or receiver end, but lower in the middle. This is because the split segments have different colors and their RC values deviate differently under lithographic variations. Due to the averaging effect across the segments, the delay deviations compensate each other and reduce overall delay variation of the interconnect. This result suggests a design guideline whereby timing-critical routes in dense patterns should preferentially receive stitches to reduce delay variation in the regime of combined CD bimodality and overlay error.

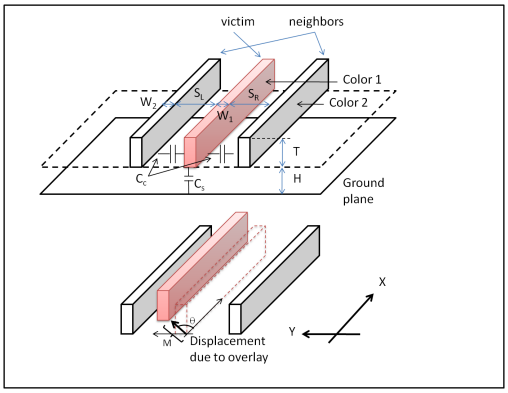
4.3.1 Resistance and Capacitance Variation Model

Ghaida and Gupta [79] study the impact of overlay on parasitic RC but they do not clarify the impact of stitching location. Here, we study RC variation of VLSI interconnects with layout configurations as illustrated in Figure 4.25. For each layout configuration, we define an interconnect under test as the victim and other interconnects as neighbors. T, $W_{1,2}$, $S_{L,R}$ and H are respectively the thickness, width, spacing and dielectric thickness of the interconnects. C_s and C_c are ground capacitance and coupling capacitance of the victim. Displacement between interconnects with different colors is modeled as a vector (M, θ) in polar coordinates, where M is magnitude and θ is polar angle of the displacement. To account for CD variation, we define

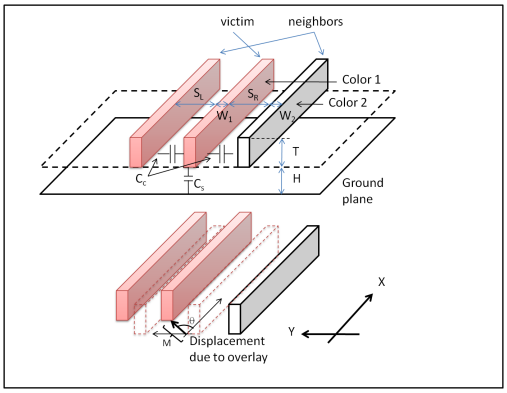
interconnect width and spacing as follows.

$$
\begin{aligned}
&W_1 = W_0 + \Delta W_1, \quad W_2 = W_0 + \Delta W_2 \\
&S_R = S_0 - 0.5(\Delta W_1) - 0.5(\Delta W_2) - M \cdot cos\ \theta \\
&S_L = \begin{cases} S_0 - 0.5(\Delta W_1) - 0.5(\Delta W_2) + M \cdot cos\ \theta & \text{for case (a) and (c) of Figure 4.25} \\ S_0 - (\Delta W_1) & \text{for case (b) of Figure 4.25} \end{cases}
\end{aligned} \tag{4.50}
$$

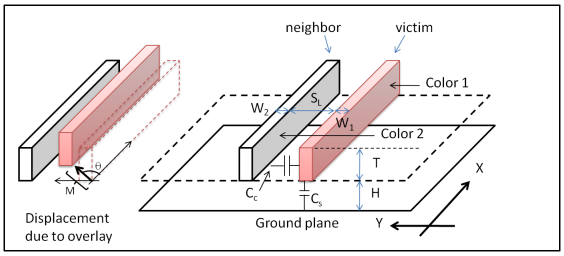
where ΔW_1 and ΔW_2 are width variations due to two independent CD distributions in DPL, W_0 is nominal width, and S_0 is nominal spacing. We model ΔW_1, ΔW_2 and M as Gaussian distributions, and θ as a uniform distribution from 0 to 2π. The values of nominal geometric dimensions and lithographic variation parameters are summarized in Table 4.13. $\mathbb{E}(\cdot)$ and $\mathbb{V}(\cdot)$ are the mean and variance functions, respectively. It should be noted that S_{min} in Table 4.13 corresponds to the minimum spacing achievable between different-color segments. Since the spacing requirements are different among the interconnect cases in Figure 4.25(a)[61], we use $S_0 = 2S_{min}$ for all interconnect cases to enable a fair comparison.



(a) Three interconnects with symmetric coloring.



(b) Three interconnects with asymmetric coloring.



(c) Two interconnects.

Figure 4.25: Interconnect dimensions and displacement due to overlay.

[61] A larger minimum spacing must exist between same-color segments as in the asymmetric case (b) of Figure 4.25.

Table 4.13: Geometric dimensions and lithographic variation parameters for $45nm$ (commercial) and $22nm$ (ITRS [232]) technologies. $\mathbb{E}(\cdot)$ and $\mathbb{V}(\cdot)$ are the mean and variance functions, respectively.

Parameter	Unit	$45nm$	$22nm$	Parameter	Unit	$45nm$	$22nm$
S_{min}	nm	70	32	$\mathbb{E}(W_1)$	nm	0	0
S_0	nm	140	64	$\mathbb{E}(W_2)$	nm	0	0
W_0	nm	70	32	$\mathbb{E}(M)$	nm	0	0
T	nm	140	60	$\mathbb{V}(W_1)$	nm^2	21.78	4.55
H	nm	140	60	$\mathbb{V}(W_2)$	nm^2	21.78	4.55
ϵ_{eff}	-	3.3	2.75	$\mathbb{V}(M)$	nm^2	21.78	4.55
ρ	Ω	27×10^9	50×10^9				

In our study, we assume that the interconnects in Figure 4.25 have stitch locations as defined in Figure 4.26, where x_1 and x_2 are lengths of victim interconnects with Color 1 and Color 2, respectively. Although interconnects with different colors will overlap at stitching locations, the overlap length is much smaller than the interconnect length (e.g., $30nm$ out of 50,000nm). Hence, we do not separately model the parasitic RC of the overlapping region.

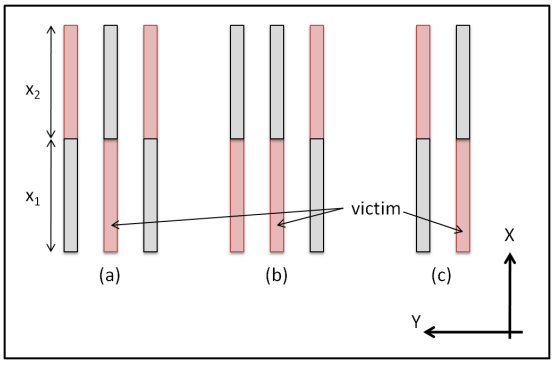


Figure 4.26: Top view of interconnect configurations from Figure 4.25, with stitches.

Capacitance Formulas for Three Parallel Interconnects with Symmetric Coloring

We use formulas from Chang [41] for ground capacitance (C_s) and from Sakurai et al. [181] for coupling capacitance (C_c). Then, total capacitance (C_v) for three parallel interconnects

with symmetric coloring (Figure 4.25(a)) is given as follows.

$$
\begin{aligned}
C_v &= (C_s + C_c) \\
C_s &= \epsilon_{ox}\left[\frac{x_1}{x_1+x_2}\cdot h(W_1) + \frac{x_2}{x_1+x_2}\cdot h(W_2)\right] \\
C_c &= \epsilon_{ox}\left[\frac{x_1}{x_1+x_2}\cdot f(W_1)\cdot g(W_1,W_2,Y) + \frac{x_2}{x_1+x_2}\cdot f(W_2)\right. \\
&\qquad \left.\cdot\, g(W_2,W_1,-Y)\right]
\end{aligned}
$$

$$
\begin{aligned}
h(W_1) &= k_1 + k_2\left(\frac{W_1}{H}\right) + k_3\left(\frac{W_1}{H}\right)^{m_1} + k_4\left(\frac{T}{H}\right)^{m_2} \\
f(W_1) &= k_5\left(\frac{W_1}{H}\right) + k_6\left(\frac{T}{H}\right) + k_7\left(\frac{T}{H}\right)^{m_3} \\
g(W_1,W_2,Y) &= \left[\left(\frac{S_0 - 0.5(W_1-W_0) - 0.5(W_2-W_0) - Y}{H}\right)^{m_4} + \right. \\
&\qquad \left.\left(\frac{S_0 - 0.5(W_1-W_0) - 0.5(W_2-W_0) + Y}{H}\right)^{m_4}\right] \\
Y &= M{\cdot}cos\,\theta, \quad \epsilon_{ox} = \epsilon_{eff}\cdot\epsilon_0
\end{aligned}
\tag{4.51}
$$

In the above, $k_1, ..., k_7$ and m_1, ..., m_4 are unitless constants, ϵ_{eff} is dielectric constant, and ϵ_0 is free-space permittivity. Values of these parameters are summarized in Table 4.13 and Table 4.14. To derive the impact of dimensional variations, we linearize the capacitance formulas using first-order Taylor series expansion[62].

$$
\begin{aligned}
C_s &\approx C_s|_{W=W_0,\ S=S_0} + \frac{\partial C_s}{\partial W_1}\left(\Delta W_1\right) \\
C_c &\approx C_c|_{W=W_0,\ S=S_0} + \frac{\partial C_c}{\partial W_1}\left(\Delta W_1\right) + \frac{\partial C_c}{\partial W_2}\left(\Delta W_2\right) + \frac{\partial C_c}{\partial Y}\left(\Delta Y\right)
\end{aligned}
\tag{4.52}
$$

Since C_v is a linear function of W_1, W_2 and Y, we calculate the mean and variance of C_v in Equation 4.53. We use subscript dpl to indicate that the capacitance is derived for double-

[62] Detailed derivations for $\frac{\partial C_s}{\partial W_1}$, $\frac{\partial C_c}{\partial W_1}$, $\frac{\partial C_c}{\partial W_2}$ and $\frac{\partial C_c}{\partial Y}$ are given in Appendix C.

patterning lithography.

$$\begin{aligned}
\mathbb{E}(C_v) &= C_s|_{W=W_0,\, S=S_0} + C_c|_{W=W_0,\, S=S_0} \\
\mathbb{V}(C_{v_dpl}) &= \left[\left(\frac{\partial C_s}{\partial W_1}\right) + \left(\frac{\partial C_c}{\partial W_1}\right)\right]^2 \cdot \mathbb{V}(W_1) + \left(\frac{\partial C_c}{\partial W_2}\right)^2 \mathbb{V}(W_2) \\
&\quad + \left(\frac{\partial C_c}{\partial Y}\right)^2 \mathbb{V}(Y) \\
\mathbb{V}(Y) &= \mathbb{E}(Y^2) - \left[\mathbb{E}(Y)\right]^2 \\
&= \sigma_M^2 \int_0^{2\pi} \frac{cos^2\theta}{2\pi} d\theta = \frac{\sigma_M^2}{2}
\end{aligned} \tag{4.53}$$

Table 4.14: Capacitance model parameters [41] [181].

Parameters	Values	Parameters	Values
m_1	0.250	k_2	1.000
m_2	0.500	k_3	1.060
m_3	0.222	k_4	1.060
m_4	-1.340	k_5	0.030
ϵ_0	$8.854F \cdot m^{-1}$	k_6	0.830
k_1	0.770	k_7	-0.070

In the case of conventional SPL all interconnects have identical widths and there is no variability due to overlay. Therefore, W_1 and W_2 are fully correlated, and Y = 0. The capacitance variance for SPL is given as follows, with a subscript spl to indicate the capacitance is for *single-patterning lithography* (SPL).

$$\mathbb{V}(C_{v\_spl}) = \left[\left(\frac{\partial C_s}{\partial W_1}\right) + \left(\frac{\partial C_c}{\partial W_1}\right) + \left(\frac{\partial C_c}{\partial W_2}\right)\right]^2 \cdot \mathbb{V}(W_1) \tag{4.54}$$

Capacitance Formulas for Three Parallel Interconnects with Asymmetric Coloring

The capacitance formulas for the asymmetric case are as follows (see Appendix D for detailed derivation).

$$\begin{aligned} g_{asym}(W_1, W_2, Y) &= \left[\left(\frac{S_0 - (W_1 - W_0)}{H}\right)^{m_4} + \right. \\ &\qquad \left.\left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) + Y}{H}\right)^{m_4}\right] \\ C_{s\_asym} &= C_s \\ C_{c\_asym} &= \epsilon_{ox} \cdot \left[\frac{x_1}{x_1 + x_2} \cdot f(W_1) \cdot g_{asym}(W_1, W_2, Y) \right. \\ &\qquad \left. + \frac{x_2}{x_1 + x_2} \cdot f(W_2) \cdot g_{asym}(W_2, W_1, -Y)\right] \end{aligned} \tag{4.55}$$

$$\begin{aligned} \mathbb{E}(C_{v\_asym}) &= C_{s\_asym}|_{W=W_0,\, S=S_0} + C_{c\_asym}|_{W=W_0,\, S=S_0} \\ \mathbb{V}(C_{v\_asym}) &= \left[\left(\frac{\partial C_{s\_asym}}{\partial W_1}\right) + \left(\frac{\partial C_{c\_asym}}{\partial W_1}\right)\right]^2 \cdot \mathbb{V}(W_1) \\ &\qquad + \left(\frac{\partial C_{c\_asym}}{\partial W_2}\right) \cdot \mathbb{V}(W_2) + \left(\frac{\partial C_{c\_asym}}{\partial W_2}\right) \cdot \mathbb{V}(Y) \end{aligned} \tag{4.56}$$

Note that the form of the Equation (4.56) is not changed compared to Equation (4.53), but we label all the terms with *asym* to indicate that the parameters are different from those in the symmetric interconnect case. Based on the equations, victim interconnects in Figure 4.25(a) and Figure 4.25(b) have the same mean capacitance value (if $S_L = S_R$) but different variations.

Capacitance Formulas for Two Parallel Interconnects

Capacitance of the victim in Figure 4.25(c) is different from that in Figure 4.25(a) as there is no right-hand side neighbor. Capacitance for two parallel interconnects (see Appendix E for detailed derivation) is given as follows, with a subscript *dual* to indicate there are two parallel

interconnects.

$$
\begin{aligned}
C_{s\_dual} &= \epsilon_{ox} \cdot \left[\frac{x_1}{x_1 + x_2} \cdot h(W_1) + \frac{x_2}{x_1 + x_2} \cdot h(W_2) \right] \\
C_{c\_dual} &= \epsilon_{ox} \cdot \left[\frac{x_1}{x_1 + x_2} \cdot f(W_1) \cdot g_{dual}(W_1, W_2, Y) \right. \\
&\qquad \left. + \frac{x_2}{x_1 + x_2} \cdot f(W_2) \cdot g_{dual}(W_2, W_1, -Y) \right] \\
g_{dual}(W_1, W_2, Y) &= \left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) - Y}{H} \right)^{m_4}
\end{aligned}
\tag{4.57}
$$

where C_{s_dual}, and C_{c_dual} are ground and coupling capacitance, respectively. Consequently, the linearized expressions for capacitance mean and variance of two parallel interconnects are as follows.

$$
\begin{aligned}
C_{s\_dual} &\approx C_{s\_dual}|_{W=W_0,\, S=S_0} + \frac{\partial C_{s\_dual}}{\partial W_1} \left(\Delta W_1 \right) \\
C_{c\_dual} &\approx C_{c\_dual}|_{W=W_0,\, S=S_0} + \frac{1}{2} \left[\frac{\partial C_{c\_dual}}{\partial W_1} \left(\Delta W_1 \right) \right. \\
&\qquad \left. + \frac{\partial C_{c\_dual}}{\partial W_2} \left(\Delta W_2 \right) + \frac{\partial C_{c\_dual}}{\partial Y} \left(\Delta Y \right) \right] \\
\mathbb{E}(C_{v\_dual}) &= C_{s\_dual}|_{W=W_0,\, S=S_0} + C_{c\_dual}|_{W=W_0,\, S=S_0} \\
\mathbb{V}(C_{v\_dual\_dpl}) &= \left[\left(\frac{\partial C_{s\_dual}}{\partial W_1} \right) + \left(\frac{\partial C_{c\_dual}}{\partial W_1} \right) \right]^2 \cdot \mathbb{V}(W_1) \\
&\qquad + \left(\frac{\partial C_{c\_dual}}{\partial W_2} \right)^2 \cdot \mathbb{V}(W_2) + \left(\frac{\partial C_{c\_dual}}{\partial Y} \right)^2 \cdot \mathbb{V}(Y) \\
\mathbb{V}(C_{v\_dual\_spl}) &= \left[\left(\frac{\partial C_{s\_dual}}{\partial W_1} \right) + \left(\frac{\partial C_{c\_dual}}{\partial W_1} \right) + \left(\frac{\partial C_{c\_dual}}{\partial W_2} \right) \right]^2 \cdot \mathbb{V}(W_1)
\end{aligned}
\tag{4.58}
$$

Interconnect Resistance Formulas

Resistance variation on the victim interconnect is only affected by the width of the victim. Therefore, all interconnect segments in Figure 4.26 have the same parasitic resistance model.

$$
\begin{aligned}
R_v &= \frac{\rho}{W \cdot T} \\
\mathbb{E}(R_v) &= R_v|_{W_1=W_0} \\
\mathbb{V}(R_v) &\approx \left(\frac{-\rho}{T \cdot W_0^2} \right)^2 \cdot \mathbb{V}(W_1)
\end{aligned}
\tag{4.59}
$$

4.3.2 Experimental Results

RC Variation Analysis

Based on the RC equations in Section 4.3.1, we calculate capacitance values of interconnects in Figure 4.25 (i.e., there is no stitching, and each victim interconnect is assigned to a single color). To compare different interconnect cases, we use $S_0 = 2S_{min}$ such that all interconnect patterns satisfy the minimum coloring spacing. Results in Table 4.15[63] show that capacitance variation with DPL is marginally smaller than with SPL. This is because the width and spacing variations of DPL interconnects are not correlated, as a consequence of the two independent exposures. Even though DPL interconnects are affected by overlay, the overall variation is less than the SPL case. As one would expect, the two-line interconnect pattern has smaller variance (relative to mean) than the three-line interconnect pattern. This is because the victim in the two-line interconnect pattern has less coupling capacitance that is sensitive to width or spacing variation (ground capacitance is identical for all interconnect patterns).

Table 4.15: Capacitance values of victim interconnects in Figure 4.25.

	$22nm$ technology					$45nm$ technology				
	3 lines			2 lines		3 lines			2 lines	
	SPL	DPL sym	DPL asym	SPL	DPL	SPL	DPL sym	DPL asym	SPL	DPL
$\mu\ (aF/\mu m)$	114.3	114.3	114.3	96.9	96.9	139.4	139.4	139.4	116.8	116.8
$3\sigma\ (aF/\mu m)$	25.5	19.8	22.8	18.3	16.0	31.4	24.1	28.0	22.0	19.2
$\frac{3\sigma}{\mu}\ (\%)$	22.3	17.3	19.9	18.9	16.5	22.5	17.3	20.1	18.9	16.4

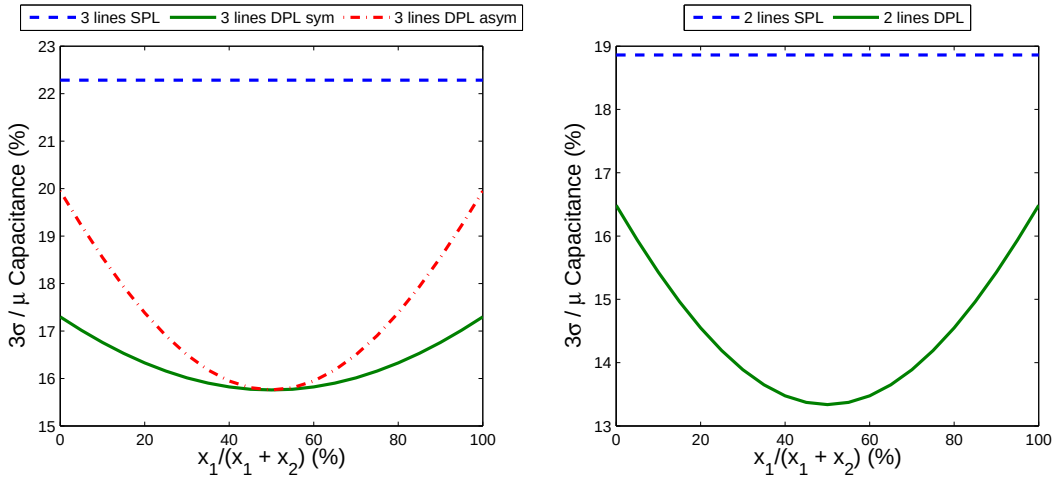
To study the effect of interconnect coloring and stitching location, we sweep the stitching location along the x-axis in Figure 4.26. Results in Figure 4.27 show that capacitance variation of DPL interconnect changes according to the stitching location. The minimal capacitance variation is achieved when stitching point is at the middle of interconnect (i.e., $x_1 = x_2$). These data suggest that

1. DPL interconnects always have lower capacitance variation (relative to mean) than SPL interconnects.

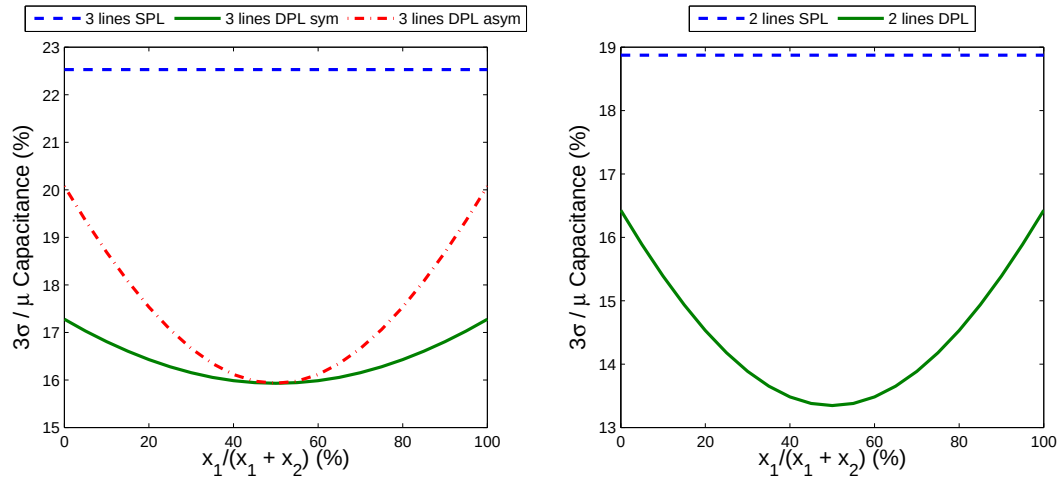
2. Redundant stitching in DPL is beneficial as it reduces capacitance variations compared to DPL with no stitching, i.e., $3\sigma/\mu$ capacitance of DPL interconnects reduce as x_1 changes

[63] Jeong et al. [96] obtained capacitance values similar to the ones in Table 4.15 using a commercial 3D RC field solver tool (*Synopsys Raphael* [255]).

from 0 (100% Color 2) to $x_1 = x_2$ (one stitch, Color 1 and Color 2 interconnect lengths are balanced.).



(a) $22nm$ technology



(b) $45nm$ technology

Figure 4.27: Capacitance variation of interconnects. Normal (SPL) interconnects have no stitching, hence their capacitance values do not vary with stitching location. Capacitance variation for DPL interconnects is minimized when the stitching point is located at the middle of the interconnect.

Delay Variation Analysis

To study the impact of stitching on circuit delay, we simulate a testing circuit illustrated in Figure 4.28. The testing circuit consists of a pair of inverters connected by a series of RC modules, each of which represents 5% of a victim interconnect (i.e., the victim interconnect is

divided into 20 identical segments). The RC values for a given RC module are calculated using the analytical equations in Section 4.3.1, with dimensions according to its color assignment. In this study, the inverter cell is obtained from a commercial library ($45nm$) and predictive technology model [219] ($22nm$). The test circuit is simulated using *Synopsys HSPICE* [251] with a $50ps$ ramp input signal and a Monte Carlo setup with 3000 trials.[64] The size of the inverter is scaled according to the length of interconnect, e.g., a $100\mu m$ interconnect uses a ($1\times$) inverter while a $1000\mu m$ interconnect uses a ($10\times$) inverter.

Figure 4.29 shows the impact of stitching location on circuit delay. Stitching location is denoted by an index from 1 to 21 which corresponds to equally-spaced discrete locations from source to sink. In particular, stitching location = 1 (resp. = 21) means that the stitching location is immediately after the driver (resp. immediately before the receiver), and the entire interconnect is assigned to Color 2 (resp. Color 1). If the stitching location = 11, the driver-side half is Color 1 and the receiver-side half is Color 2.

All testcases in Figure 4.29 show that DPL interconnect has less delay variation compared to the SPL case. As mentioned earlier, this is due the averaging effect of DPL interconnects. We also notice that stitching around the middle of interconnect leads to minimal delay variation (long interconnect). This is expected because the capacitance variation of interconnect is minimal when the portions of Color 1 and Color 2 are equal (for DPL). Note that for all testcases, minimum $3\sigma/mean$ is attained when stitching location is slightly shifted towards the driver side. This is because circuit delay is more sensitive to RC changes on the driver side, due to the resistance shielding effect. Resistance shielding implies that driver-side capacitance has more contribution to RC delay than receiver-side capacitance. As a result, the stitching location shifts slightly toward the driver side to balance the effective RC of interconnects with Color 1 and Color 2.

To model the bimodal distribution in DPL, we perturb the mean of interconnect Color 1 by $\pm 2nm$. Figures 4.29(c) and (d) show that testcases with $\pm 2nm$ ΔCD mean behave similarly to those with ΔCD mean = 0. In other words, the impact of ΔCD mean is negligible for circuit delay analysis. Similarly, the delay variation trends for $45nm$ and $22nm$ technologies (Figures 4.29(b) and (e)) are qualitatively the same. This hints that we should expect similar delay variation phenomena in future technologies. Comparing Figure 4.29(a) and Figure 4.29(b), we see that the impact of stitching location on short interconnect ($100\mu m$) is slightly less than that on long interconnects ($1000\mu m$), but the trends are similar.

[64] Small sample size for Color 1 and Color 2 CD random variables can induce additional (unwanted) mean CD shift between them. This may lead to incorrect interpretations on the impact of stitching.

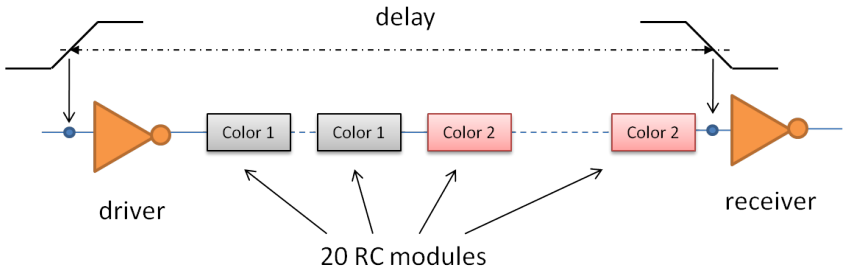


Figure 4.28: Circuit to used study the impact of stitching locations. Each of the 20 RC modules represents 5% of the parasitic RC of the entire interconnect, and is assigned to either Color 1 or Color 2. There is only one splitting/stitching point along the modules.

4.3.3 Conclusions

We derive analytical RC equations for LELE DPL-based interconnects to study the impact of stitching insertion on interconnects RC and circuit performance. Our experimental results show that DPL without any stitching along victim interconnect has less delay variation compared to SPL. This suggests that layout decomposition algorithms should alternate color assignments of interconnects, such that delay variation is partially alleviated due to averaging effect. The results also show that interconnect with a stitch always has a smaller delay variation compared to interconnect without any stitch. This implies that, long interconnect should preferentially receive stitches to reduce delay variation. The results in Figure 4.29 suggest that stitching location should be placed along an interconnect such that the interconnect has equal proportion of segments. Although the stitching location is slightly shifted towards driver side, there is only a small difference between the minimum delay variation versus the case where a stitch is placed at the middle of the interconnect. Therefore, always splitting an interconnect at its midpoint could be a simple yet near-optimal mask optimization strategy for minimum performance variation.

4.3.4 Appendix C: Symmetric 3-lines Interconnect

$$
\begin{aligned}
\text{let } h(W_1) &= k_1 + k_2\left(\frac{W_1}{H}\right) + k_3\left(\frac{W_1}{H}\right)^{m_1} + k_4\left(\frac{T}{H}\right)^{m_2} \\
f(W_1) &= k_5\left(\frac{W_1}{H}\right) + k_6\left(\frac{T}{H}\right) + k_7\left(\frac{T}{H}\right)^{m_3} \\
g(W_1, W_2, Y) &= \left[\left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) - Y}{H}\right)^{m_4}\right. \\
&\quad \left. + \left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) + Y}{H}\right)^{m_4}\right] \\
C_s &= \epsilon_{ox} \cdot \left[\frac{x_1}{x_1 + x_2} \cdot h(W_1) + \frac{x_2}{x_1 + x_2} \cdot h(W_2)\right] \\
C_c &= \epsilon_{ox} \cdot \left[\frac{x_1}{x_1 + x_2} \cdot f(W_1) \cdot g(W_1, W_2, Y) + \frac{x_2}{x_1 + x_2} \cdot f(W_2) \cdot g(W_2, W_1, -Y)\right] \\
\frac{\partial h(W_1)}{\partial W_1} &= \frac{k_2}{H} + \frac{k_3 \cdot m_1 \cdot W_1^{(m_1 - 1)}}{H^{m_1}} \\
\frac{\partial h(W_2)}{\partial W_2} &= \frac{k_2}{H} + \frac{k_3 \cdot m_1 \cdot W_2^{(m_1 - 1)}}{H^{m_1}} \\
\frac{\partial f(W_1)}{\partial W_1} &= \frac{\partial f(W_2)}{\partial W_2} = \frac{k_5}{H}, \quad \frac{\partial f(W_1)}{\partial W_2} = \frac{\partial f(W_2)}{\partial W_1} = 0 \\
\frac{\partial g(W_1, W_2, Y)}{\partial W_1} &= \frac{\partial g(W_1, W_2, Y)}{\partial W_2} \\
&= \frac{-m_4}{2H}\left[\left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) - Y}{H}\right)^{m_4 - 1}\right. \\
&\quad \left. + \left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) + Y}{H}\right)^{m_4 - 1}\right] \\
\frac{\partial g(W_2, W_1, -Y)}{\partial W_1} &= \frac{\partial g(W_2, W_1, -Y)}{\partial W_2} \\
&= \frac{-m_4}{2H}\left[\left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) + Y}{H}\right)^{m_4 - 1}\right. \\
&\quad \left. + \left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) - Y}{H}\right)^{m_4 - 1}\right] \\
\frac{\partial g(W_1, W_2, Y)}{\partial Y} &= \frac{\partial g(W_2, W_1, -Y)}{\partial Y} \\
&= \frac{-m_4}{H}\left[\left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) - Y}{H}\right)^{m_4 - 1}\right. \\
&\quad \left. - \left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) + Y}{H}\right)^{m_4 - 1}\right] \\
\frac{\partial C_s}{\partial W_1} &= \epsilon_{ox} \cdot \frac{x_1}{x_1 + x_2} \cdot \frac{\partial h(W_1)}{\partial W_1}, \quad \frac{\partial C_s}{\partial W_2} = \epsilon_{ox} \cdot \frac{x_2}{x_1 + x_2} \cdot \frac{\partial h(W_2)}{\partial W_2} \\
\frac{\partial C_c}{\partial W_1} &= \epsilon_{ox} \cdot \left[\frac{x_1}{x_1 + x_2}\left(\frac{\partial f(W_1)}{\partial W_1} \cdot g(W_1, W_2, Y) + \frac{\partial g(W_1, W_2, Y)}{\partial W_1} \cdot f(W_1)\right)\right. \\
&\quad \left. + \frac{x_2}{x_1 + x_2}\left(\frac{\partial g(W_2, W_1, -Y)}{\partial W_1} \cdot f(W_2)\right)\right] \\
\frac{\partial C_c}{\partial W_2} &= \epsilon_{ox} \cdot \left[\frac{x_1}{x_1 + x_2}\left(\frac{\partial g(W_1, W_2, Y)}{\partial W_2} \cdot f(W_1)\right)\right. \\
&\quad \left. + \frac{x_2}{x_1 + x_2}\left(\frac{\partial f(W_2)}{\partial W_2} \cdot g(W_1, W_2, Y) + \frac{\partial g(W_2, W_1, -Y)}{\partial W_2} \cdot f(W_2)\right)\right] \\
\frac{\partial C_c}{\partial Y} &= \epsilon_{ox} \cdot \left[\frac{x_1}{x_1 + x_2}\left(f(W_1) \cdot \frac{\partial g(W_1, W_2, Y)}{\partial Y}\right) + \frac{x_2}{x_1 + x_2}\left(f(W_2) \cdot \frac{\partial g(W_2, W_1, -Y)}{\partial Y}\right)\right]
\end{aligned}
$$

4.3.5 Appendix D: Asymmetric 3-lines Interconnect

$$
\begin{aligned}
g_{asym}(W_1,W_2,Y) &= \left[\left(\frac{S_0-(W_1-W_0)}{H}\right)^{m_4}+\left(\frac{S_0-0.5(W_1-W_0)-0.5(W_2-W_0)+Y}{H}\right)^{m_4}\right]\\
C_{s\_asym} &= C_s\\
C_{c\_asym} &= \epsilon_{ox}\cdot\left[\frac{x_1}{x_1+x_2}\cdot f(W_1)\cdot g_{asym}(W_1,W_2,Y)\right.\\
&\quad\left.+\frac{x_2}{x_1+x_2}\cdot f(W_2)\cdot g_{asym}(W_2,W_1,-Y)\right]\\
\frac{\partial g_{asym}(W_1,W_2,Y)}{\partial W_1} &= \frac{-m_4}{2H}\left[2\left(\frac{S_0-(W_1-W_0)}{H}\right)^{m_4-1}+\left(\frac{S_0-0.5(W_1-W_0)-0.5(W_2-W_0)+Y}{H}\right)^{m_4-1}\right]\\
\frac{\partial g_{asym}(W_1,W_2,Y)}{\partial W_2} &= \frac{-m_4}{2H}\left(\frac{S_0-0.5(W_1-W_0)-0.5(W_2-W_0)+Y}{H}\right)^{m_4-1}\\
\frac{\partial g_{asym}(W_2,W_1,-Y)}{\partial W_1} &= \frac{-m_4}{2H}\left(\frac{S_0-0.5(W_1-W_0)-0.5(W_2-W_0)-Y}{H}\right)^{m_4-1}\\
\frac{\partial g_{asym}(W_2,W_1,-Y)}{\partial W_2} &= \frac{-m_4}{2H}\left[2\left(\frac{S_0-(W_2-W_0)}{H}\right)^{m_4-1}+\left(\frac{S_0-0.5(W_1-W_0)-0.5(W_2-W_0)-Y}{H}\right)^{m_4-1}\right]\\
\frac{\partial g_{asym}(W_1,W_2,Y)}{\partial Y} &= \frac{m_4}{H}\left[\left(\frac{S_0-0.5(W_1-W_0)-0.5(W_2-W_0)+Y}{H}\right)^{m_4-1}\right]\\
\frac{\partial g_{asym}(W_2,W_1,-Y)}{\partial Y} &= \frac{-m_4}{H}\left[\left(\frac{S_0-0.5(W_1-W_0)-0.5(W_2-W_0)-Y}{H}\right)^{m_4-1}\right]\\
\frac{\partial C_{s\_asym}}{\partial W_1} &= \frac{\partial C_s}{\partial W_1}\\
\frac{\partial C_{s\_asym}}{\partial W_2} &= \frac{\partial C_s}{\partial W_2}\\
\frac{\partial C_{c\_asym}}{\partial W_1} &= \epsilon_{ox}\cdot\left[\frac{x_1}{x_1+x_2}\left(\frac{\partial f(W_1)}{\partial W_1}\cdot g_{asym}(W_1,W_2,Y)+\frac{\partial g_{asym}(W_1,W_2,Y)}{\partial W_1}\cdot f(W_1)\right)\right.\\
&\quad\left.+\frac{x_2}{x_1+x_2}\left(\frac{\partial g_{asym}(W_2,W_1,-Y)}{\partial W_1}\cdot f(W_2)\right)\right]\\
\frac{\partial C_{c\_asym}}{\partial W_2} &= \epsilon_{ox}\cdot\left[\frac{x_1}{x_1+x_2}\left(\frac{\partial g_{asym}(W_1,W_2,Y)}{\partial W_2}\cdot f(W_1)\right)\right.\\
&\quad\left.+\frac{x_2}{x_1+x_2}\left(\frac{\partial f(W_2)}{\partial W_2}\cdot g_{asym}(W_1,W_2,Y)+\frac{\partial g_{asym}(W_2,W_1,-Y)}{\partial W_2}\cdot f(W_2)\right)\right]\\
\frac{\partial C_{c\_asym}}{\partial Y} &= \epsilon_{ox}\cdot\left[\frac{x_1}{x_1+x_2}\left(f(W_1)\cdot\frac{\partial g_{asym}(W_1,W_2,Y)}{\partial Y}\right)\right.\\
&\quad\left.+\frac{x_2}{x_1+x_2}\left(f(W_2)\cdot\frac{\partial g_{asym}(W_2,W_1,-Y)}{\partial Y}\right)\right]
\end{aligned}
$$

4.3.6 Appendix E: Asymmetric 2-lines Interconnect

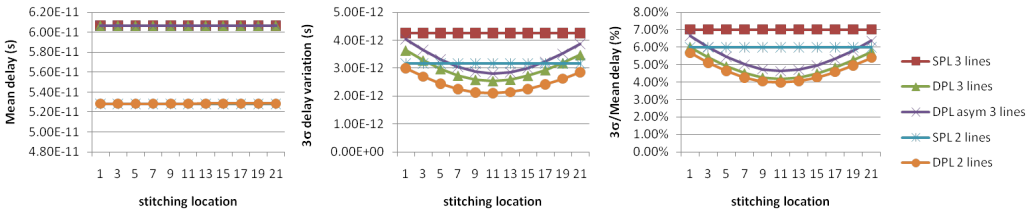
$$
\begin{aligned}
\text{let } g_{dual}(W_1, W_2, Y) &= \left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) - Y}{H}\right)^{m_4} \\
C_s &= \epsilon_{ox} \cdot \left[\frac{x_1}{x_1 + x_2} \cdot h(W_1) + \frac{x_2}{x_1 + x_2} \cdot h(W_2)\right] \\
C_c &= \epsilon_{ox} \cdot \left[\frac{x_1}{x_1 + x_2} \cdot f(W_1) \cdot g_{dual}(W_1, W_2, Y) + \frac{x_2}{x_1 + x_2} \cdot f(W_2) \cdot g_{dual}(W_2, W_1, -Y)\right] \\
\frac{\partial g_{dual}(W_1, W_2, Y)}{\partial W_1} &= \frac{\partial g_{dual}(W_1, W_2, Y)}{\partial W_2} = \frac{-m_4}{2H}\left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) - Y}{H}\right)^{m_4 - 1} \\
\frac{\partial g_{dual}(W_2, W_1, -Y)}{\partial W_1} &= \frac{\partial g_{dual}(W_2, W_1, -Y)}{\partial W_2} = \frac{-m_4}{2H}\left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) + Y}{H}\right)^{m_4 - 1} \\
\frac{\partial g_{dual}(W_1, W_2, Y)}{\partial Y} &= \frac{-m_4}{H}\left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) - Y}{H}\right)^{m_4 - 1} \\
\frac{\partial g_{dual}(W_2, W_1, -Y)}{\partial Y} &= \frac{m_4}{H}\left(\frac{S_0 - 0.5(W_1 - W_0) - 0.5(W_2 - W_0) + Y}{H}\right)^{m_4 - 1} \\
\frac{\partial C_{s\_dual}}{\partial W_1} &= \frac{\partial C_s}{\partial W_1} \\
\frac{\partial C_{s\_dual}}{\partial W_2} &= \frac{\partial C_s}{\partial W_2} \\
\frac{\partial C_c}{\partial W_1} &= \epsilon_{ox} \cdot \left[\frac{x_1}{x_1 + x_2}\left(\frac{\partial f(W_1)}{\partial W_1} \cdot g_{dual}(W_1, W_2, Y) + \frac{\partial g_{dual}(W_1, W_2, Y)}{\partial W_1} \cdot f(W_1)\right)\right. \\
&\quad \left. + \frac{x_2}{x_1 + x_2}\left(\frac{\partial g_{dual}(W_2, W_1, -Y)}{\partial W_1} \cdot f(W_2)\right)\right] \\
\frac{\partial C_c}{\partial W_2} &= \epsilon_{ox} \cdot \left[\frac{x_1}{x_1 + x_2}\left(\frac{\partial g_{dual}(W_1, W_2, Y)}{\partial W_2} \cdot f(W_1)\right)\right. \\
&\quad \left. + \frac{x_2}{x_1 + x_2}\left(\frac{\partial f(W_2)}{\partial W_2} \cdot g_{dual}(W_1, W_2, Y) + \frac{\partial g_{dual}(W_2, W_1, -Y)}{\partial W_2} \cdot f(W_2)\right)\right] \\
\frac{\partial C_c}{\partial Y} &= \epsilon_{ox} \cdot \left[\frac{x_1}{x_1 + x_2}\left(f(W_1) \cdot \frac{\partial g_{dual}(W_1, W_2, Y)}{\partial Y}\right)\right. \\
&\quad \left. + \frac{x_2}{x_1 + x_2}\left(f(W_2) \cdot \frac{\partial g_{dual}(W_2, W_1, -Y)}{\partial Y}\right)\right]
\end{aligned}
$$

4.4 Acknowledgments

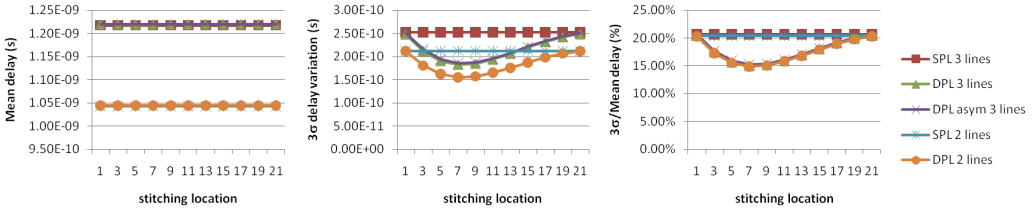
Chapter 4 is in part a reprint of "On Electrical Modeling of Imperfect Diffusion Patterning", *Proc. IEEE/ACM International Conference on VLSI Design*, 2010, "Performance and Variability Driven Guidelines for BEOL Layout Decomposition with LELE Double Patterning", *Proc. SPIE/BACUS Symposium on Photomask Technology and Management*, 2011, "Measurement and Optimization of Electrical Process Window", *SPIE Journal of Microlithography, Microfabrication and Microsystems* 10(1) (2011) "Measurement and Optimization of Electrical Process Window", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2010, "Design-Dependent Process Monitoring for Wafer Manufac-

turing and Test Cost Reduction", *IEEE Transactions on Semiconductor Manufacturing* 25(3) (2012), and "Design-Dependent Process Monitoring for Back-End Manufacturing Cost Reduction", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2010.

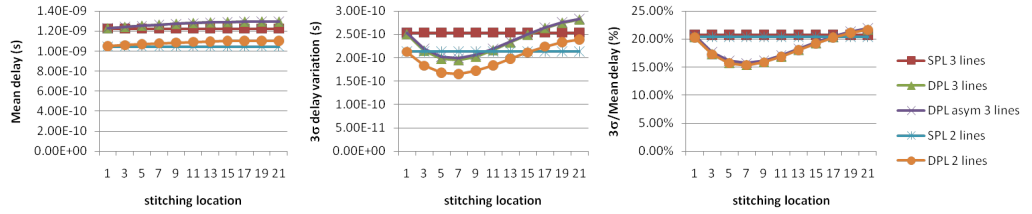
I would like to thank my co-authors Lerong Cheng, Professor Puneet Gupta, Dr. Kwangok Jeong, Abde Ali Kagalwalla, Andrew B. Kahng and Aashish Pant.



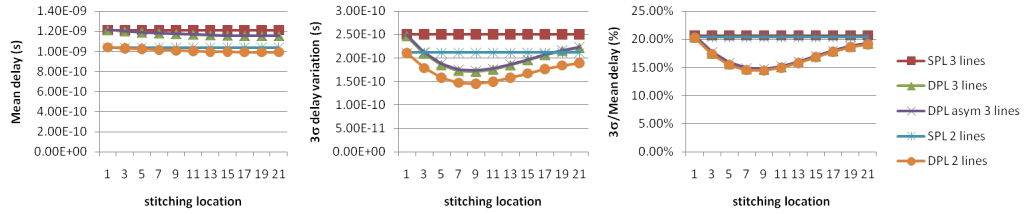
(a) $22nm$ technology, length = $100\mu m$, ΔCD mean = $0nm$, inverter size = 1X.



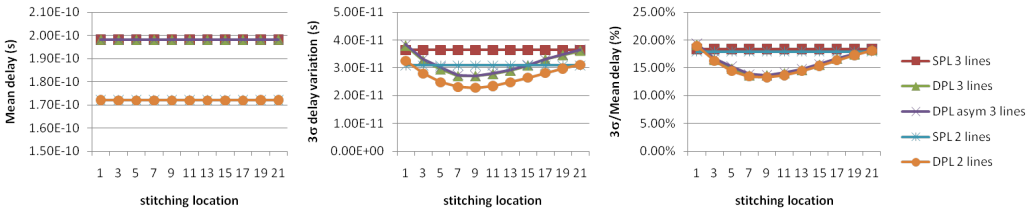
(b) $22nm$ technology, length = $1000\mu m$, ΔCD mean = $0nm$, inverter size = 10X.



(c) $22nm$ technology, length = $1000\mu m$, ΔCD mean = $-2nm$, inverter size = 10X.



(d) $22nm$ technology, length = $1000\mu m$, ΔCD mean = $2nm$, inverter size = 10X.



(e) $45nm$ technology, length = $1000\mu m$, ΔCD mean = $0nm$, inverter size = 10X.

Figure 4.29: Average delay (rising and falling transitions) of an inverter and its variation due to interconnect.

Bibliography

[1] K. Agarwal, M. Agarwal, D. Sylvester and D. Blauuw, "Statistical Interconnect Metrics for Physical-Design Optimization", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 25(7) (2006), pp. 1273-1288.

[2] A. Agarwal, D. Blaauw and V. Zolotov, "Statistical Timing Analysis for Intra-Die Process Variations with Spatial Correlations", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2003, pp. 900-907.

[3] M. A. Alam, "A Critical Examination of the Mechanics of Dynamic NBTI for PMOSFETs", *Proc. IEEE International Electron Devices Meeting*, 2003, pp. 14.4.1-14.4.4.

[4] M. A. Alam, K. Roy and C. Augustine, "Reliability- and Process-Variation Aware Design of Integrated Circuits - A Broader Perspective", *Proc. IEEE International Reliability Physics Symposium*, 2011, pp. 4A.1.1-4A.1.11.

[5] C. Amin, N. Menezes, K. Killpack, F. Dartu, U. Choudhury, N. Hakim and Y. Ismail, "Statistical Static Timing Analysis: How Simple Can We Get?", *Proc. IEEE/ACM Design Automation Conference*, 2005, pp. 652-657.

[6] K. V. Arnim, C. Pacha, K. Hofmann, T. Schulz, K. Schrüfer and J. Berthold, "An Effective Switching Current Methodology to Predict the Performance of Complex Digital Circuits", *Proc. IEEE International Electron Devices Meeting*, 2007, pp. 483-486.

[7] D. Arthur and S. Vassilvitskii, "K-Means++: The Advantages of Careful Seeding", *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 2007, pp. 1027-1035.

[8] V. Axelrad, A. Shibkov, G. Hill, H.-J. Lin, C. Tabery, D. White, V. Boksha and R. Thilmany, "A Novel Design-Process Optimization Technique Based on Self-Consistent Electrical Performance Evaluation", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, Vol. 5756, 2005, pp. 419-426.

[9] S. Banerjee, P. Elakkumanan, L. W. Liebmann and M. Orshansky, "Electrically Driven Optical Proximity Correction Based on Linear Programming", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2008, pp. 473-479.

[10] M. Bashir and L. Milor, "Towards a Chip Level Reliability Simulator for Copper/Low-k Backend Processes", *Proc. Design, Automation and Test in Europe*, 2010, pp. 279-282.

[11] M. Basoglu, M. Orshansky and M. Erez, "NBTI-Aware DVFS: A New Approach to Saving Energy and Increasing Processor Lifetime", *Proc. International Symposium on Low Power Electronic Design*, 2010, pp. 253-258.

[12] R. J. Beffa, "Method in an Integrated Circuit (IC) Manufacturing Process for Identifying and Redirecting IC's Mis-Processed During Their Manufacture", *U.S. Patent* No. US6363329B2, March 2002.

[13] C. Bencher, Applied Materials, Inc., *personal communication*, July 2011.

[14] M. Berkelaar, "Statistical Delay Calculation: a Linear Time Method", *Proc. of TAU*, 1997, pp. 4-5.

[15] A. Berman, "Time-Zero Dielectric Reliability Test By a Ramp Method", *Proc. IEEE International Reliability Physics Symposium*, 1981, p. 204.

[16] J. Bhasker and R. Chadha, *Static Timing Analysis for Nanometer Designs: A Practical Approach*, Springer, 2009.

[17] M. Bhushan, A. Gattiker, M. Ketchen and K. K. Das, "Ring Oscillators for CMOS Process Tuning and Variability Control", *IEEE Transactions on Semiconductor Manufacturing* 19(1) (2006), pp. 10-18.

[18] D. Blaauw, K. Chopra, A. Srivastava and L. Scheffer, "Statistical Timing Analysis: From Basic Principles to State of the Art", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27(4) (2008), pp. 589-607.

[19] T. Black, "A Critical Path Based Parametric Ring Oscillator", *Master's Thesis*, Texas Tech University, 2000.

[20] F. Boeuf, F. Arnaud, C. Boccaccio, F. Salvetti, J. Todeschini, L. Pain, M. Jurdit, S. Manakli, B. Icard, N. Planes, N. Gierczynski, S. Denorme, B. Borot, C. Ortolland, B. Duriez, B. Tavel, P. Gouraud, M. Broekaart, V. Dejonghe, P. Brun, F. Guyader, P. Morin, C. Reddy, M. Aminpur, C. Laviron, S. Smith, J. P. Jacquemin, M. Mellier, F. André, N. Bicais-Lepinay, S. Jullian, J. Bustos and T. Skotnicki, "0.248 μm^2 and 0.334 μm^2 Conventional Bulk 6T-SRAM Bit-Cells for 45nm Node Low Cost-General Purpose Applications", *Proc. Symposium on VLSI Technology*, 2005, pp. 130-131.

[21] S. Borkar, T. Karnik, S. Narendra, J. W. Tschanz, A. Keshavarzi and V. De, "Parameter Variations and Impact on Circuits and Microarchitecture", *Proc. IEEE/ACM Design Automation Conference*, 2003, pp. 338-342.

[22] R. B. Brawhear, N. Menezes, C. Oh, L. T. Pillage and M. R. Mercer, "Predicting Circuit Performance Using Circuit-Level Statistical Timing Analysis", *Proc. Design and Test Conference*, 1994, pp. 332-337.

[23] R. Brain, S. Agrawal, D. Becher, R. Bigwood, M. Buehler, V. Chikarmane, M. Childs, J. Choi, S. Daviess, C. Ganpule, J. He, P. Hentges, I. Jin, S. Klopcic, G. Malyavantham, B. McFadden, J. Neulinger, J. Neirynck, Y. Neirynck, C. Pelto, P. Plekhanov, Y. Shusterman, T. Van, M. Weiss, S. Williams, F. Xia, P. Yashar and A. Yeoh, "Low-k Interconnect Stack

with a Novel Self-Aligned Via Patterning Process for 32nm High Volume Manufacturing", *Proc. IEEE International Interconnect Technology Conference*, 2009, pp. 249-251.

[24] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in FORTRAN", *Proc. ISCAS*, 1985, pp. 677-692.

[25] C. Bruynseraede, Zs. Tökei, F. Iacopi, G. P. Beyer, J. Michelon and K. Maex, "The Impact of Scaling on Interconnect Reliability", *Proc. IEEE International Reliability Physics Symposium*, 2005, pp. 7-17.

[26] D. Bull, S. Das, K. Shivashankar, G. S. Dasika, K. Flautner and D. Blaauw, "A Power-Efficient 32 bit ARM Processor Using Timing-Error Detection and Correction for Transient-Error Tolerance and Adaptation to PVT Variation", *Journal of Solid State Circuits* 46(1) (2011), pp. 18-31.

[27] T. D. Burd, T. A. Pering, A. J. Stratakos and R. W. Brodersen, "A Dynamic Voltage Scaled Microprocessor System", *Journal of Solid State Circuits* 35(11) (2000), pp. 1571-1580.

[28] L. M. Burns, L. Dauphinee, R. A. Gomez and J. Y. C. Chang, "Process Monitor for Monitoring and Compensating Circuit Performance", *U.S. Patent* No. US7375540B2, May 2008.

[29] Y. Cao, P. Gupta, A. B. Kahng, D. Sylvester and J. Yang, "Design Sensitivities to Variability: Extrapolation and Assessments in Nanometer VLSI", *Proc. IEEE International Conference on ASIC/SoC*, 2002, pp. 411-415.

[30] J. M. Carulli Jr., D. C. Wrobbel, A. Mehta, K. E. Krause Jr., B. E. Campbell and F. A. Valente, "Frequency Distribution Modeling for High-Speed Microprocessors Using On-Chip Ring-Oscillators", *Proc. SPIE*, Vol. 3884, 1999, pp. 146-155.

[31] T.-B. Chan, W.-T. J. Chan and A. B. Kahng, "Impact of Adaptive Voltage Scaling on Aging-Aware Signoff", *Proc. Design, Automation and Test in Europe*, 2013, pp. 1683-1688.

[32] T.-B. Chan, R. S. Ghaida and P. Gupta, "Electrical Modeling of Lithographic Imperfections", *Proc. IEEE/ACM International Conference on VLSI Design*, 2010, pp. 423-428.

[33] T.-B. Chan and P. Gupta, "On Electrical Modeling of Imperfect Diffusion Patterning", *Proc. IEEE/ACM International Conference on VLSI Design*, 2010, pp. 224-229.

[34] T.-B. Chan, P. Gupta, A. Kahng and L. Lai, "DDRO: A Novel Performance Monitoring Methodology Based on Design-Dependent Ring Oscillators", *Proc. International Symposium on Quality Electronic Design*, 2012, pp. 633-640.

[35] T.-B. Chan, K. Jeong and Andrew B. Kahng, "Performance and Variability Driven Guidelines for BEOL Layout Decomposition with LELE Double Patterning", *Proc. SPIE/BACUS Symposium on Photomask Technology and Management*, 2011, Vol. 8166, pp. 81663O-1-81663O-12.

[36] T.-B. Chan and A. B. Kahng, "Tunable Sensors for Process-Aware Voltage Scaling", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2012, pp. 7-14.

[37] T.-B. Chan, A. B. Kahng, J. Li and S. Nath, "Optimization of Overdrive Signoff", *Proc. Asia and South Pacific Design Automation Conference*, 2013, pp. 344-349.

[38] T.-B. Chan, A. Pant, L. Cheng and P. Gupta, "Design Dependent Process Monitoring for Back-End Manufacturing Cost Reduction", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2010, pp. 116-122.

[39] T.-B. Chan, J. Sartori, P. Gupta and R. Kumar, "On the Efficacy of NBTI Mitigation Techniques", *Proc. Design, Automation and Test in Europe*, 2011, pp. 1-6.

[40] S. Chandra, A. Raghunathan and S. Dey, "Variation-Aware Voltage Level Selection", *IEEE Transactions on Very Large Scale Integration Systems* 20(5) (2012), pp. 925-936.

[41] W. H. Chang, "Analytic IC-Metal-Line Capacitance Formulas", *IEEE Transactions on Microwave Theory* MTT-24(2) (1976), pp. 608-611.

[42] C. T. Chang and H. L. Chang, "Improving TDDB Reliability in Cu Damascene by Modulating ESL Structure", *Proc. IEEE International Interconnect Technology Conference*, 2012, pp. 1-3.

[43] H. Chang and S. Sapatnekar, "Statistical Timing Analysis Considering Spatial Correlations Using a Single PERT-Like Traversal", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2003, pp. 621-625.

[44] F. Chen, O. Bravo, K. Chanda, P. McLaughlin, T. Sullivan, J. Goill, J. Lloyd, F. Kontra and J. Aitken, "Comprehensive Study of Low-k SiCOH TDDB Phenomena and Its Reliability Lifetime Model Development", *Proc. IEEE International Reliability Physics Symposium*, 2006, p. 46.

[45] M. Chen and A. Orailoglu, "Test Cost Minimization Through Adaptive Test Development", *Proc. IEEE International Conference on Computer Design*, 2008, pp. 234-239.

[46] X. Chen and L.-S. Peh, "Leakage Power Modeling and Optimization in Interconnetion Networks", *Proc. International Symposium on Low Power Electronic Design*, 2003, pp. 90-95.

[47] F. Chen, M. A. Shinosky and J. M. Aitken, "Extreme-Value Statistics and Poisson Area Scaling with a Fatal-Area Ratio for Low-k Dielectric TDDB Modeling", *IEEE Transactions on Electron Devices* 58(9) (2011), pp. 3089-3098.

[48] X. Chen, Y. Wang, Y. Cao, Y. Ma and H. Yang, "Variation-Aware Supply Voltage Assignment for Simultaneous Power and Aging Optimization", *IEEE Transactions on Very Large Scale Integration Systems* 20(11) (2012), pp. 2143-2147.

[49] L. Cheng, P. Gupta, K. Qian, C. Spanos and L. He, "Physically Justifiable Die-Level Modeling of Spatial Variation in View of Systematic Across Wafer Variability", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30(3) (2011), pp. 388-401.

[50] E. Chery, X. Federspiel, G. Beylier, C. Besset and D. Roy, "Back-End Dielectrics Reliability Under Unipolar and Bipolar AC-Stress", *Proc. IEEE International Reliability Physics Symposium*, 2012, pp. 3A.5.1-3A.5.6.

[51] T.-B. Chiou, R. Socha, H. Chen, L. Chen, S. Hsu, P. Nikolsky, A. V. Oosten and A. C. Chen, "Development of Layout Split Algorithms and Printability Evaluation for Double Patterning Technology", *Proc. SPIE Optical Microlithography*, Vol. 6924, 2008, pp. 69243-1-69243-10.

[52] M. Cho, Y. Ban and D. Z. Pan, "Double Patterning Technology Friendly Detailed Routing", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2008, pp. 506-511.

[53] C. Y. Cho, D. D. Kim, J. H. Kim, D. Y. Lim and S. Y. Cho, "Early Prediction of Product Performance and Yield Via Technology Benchmark", *Proc. IEEE Custom Integrated Circuits Conference*, 2008, pp. 205-208.

[54] N. B. Cobb and Y. Granik, "Using OPC to Optimize for Image Slope and Improve Process Window", *Proc. SPIE*, Vol. 5130, 2003, pp. 838-846.

[55] D. Corley and H. W. Littlebury, "Integral Semiconductor Wafer Map Recording", *U.S. Patent* No. US5256578A, October 1993.

[56] K. Croes and Z. Tökei, "E- and $\sqrt{E}$-Model Too Conservative to Describe Low Field Time Dependent Dielectric Breakdown", *Proc. IEEE International Reliability Physics Symposium*, 2010, pp. 543-548.

[57] W. Dai and H. Ji, "Timing Analysis Taking Into Account Interconnect Process Variation", *IEEE International Workshop on Statistical Methodology*, 2001, pp. 51-53.

[58] B. Das, B. Amrutur, H. Jamadagni, N. Arvind and V. Visvanathan, "Within-Die Gate Delay Variability Measurement Using Reconfigurable Ring Oscillator", *IEEE Transactions on Semiconductor Manufacturing* 22(2) (2009), pp. 256-267.

[59] S. Das, D. Roberts, S. Lee, S. Pant, D. Blaauw, T. Austin, K. Flautner and T. Mudge, "A Self-Tuning DVS Processor Using Delay-Error Detection and Correction", *Proc. IEEE International Conference on Integrated Circuit Design and Technology*, 2006, pp. 792-804.

[60] S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull and D. T. Blaauw, "Razor II: In Situ Error Detection and Correlation for PVT and SER Tolerance", *Journal of Solid State Circuits* 44(1) (2009), pp. 32-48.

[61] K. K. Das, S. G. Walker and M. Bhushan, "An Integrated CAD Methodology for Evaluating Mosfet and Parasitic Extraction Models and Variabilitiy", *Proc. of the IEEE* 95(3) (2007), pp. 670-687.

[62] S. Devadas, H. Jyu, K. Keutzer and S. Malik, "Statistical Timing Analysis of Combinational Circuits", *Proc. IEEE International Conference on Computer Design*, 1992, pp. 38-43.

[63] A. Devgan and S. Nassif, "Power Variability and Its Impact on Design", *Proc. International Symposium on Quality Electronic Design*, 2005, pp. 284-290.

[64] S. Dhar, D. Maksimovic and B. Kranzen, "Closed-Loop Adaptive Voltage Scaling Controller for Standard-Cell ASICs", *Proc. International Symposium on Low Power Electronic Design*, 2008, pp. 103-107.

[65] A. Drake, R. Senger, H. Singh, G. Carpenter and N. James, "Dynamic Measurement of Critical-Path Timing", *Proc. IEEE International Conference on Integrated Circuit Design and Technology and Tutorial*, 2008, pp. 249-252.

[66] Y. Du, Q. Ma, H. Song, J. Shiely, G. Luk-Pat, A. Miloslavsky and M. D. F. Wong, "Spacer-is-Dielectric-Compliant Detailed Routing for Self-Aligned Double Patterning Lithography", *Proc. IEEE/ACM Design Automation Conference*, 2013, pp. 1-6.

[67] M. Dusa, J. Quaedackers, O. F. A. Larsen, J. Meessen, E. V. D. Heijden, G. Dicker, O. Wismans, P. D. Haas, K. V. I. Schenau, J. Finders, B. Vleeming, G. Storms, P. Jaenen, S. Cheng and M. Maenhoudt, "Pitch Doubling Through Dual-Patterning Lithography: Challenges in Integration and Litho Budgets", *SPIE Optical Microlithography*, Vol. 6520, 2007, pp. 65200G-1-65200G-10.

[68] M. Elgebaly, K. Z. Malik, L. G. Chua-Eoan and S. Jung, "Adaptive Voltage Scaling for an Electronics Device", *U.S. Patent* No. 7417482, August 2008.

[69] M. Elgebaly and M. Sachdev, "Variation-Aware Adaptive Voltage Scaling System", *IEEE Transactions on Very Large Scale Integration Systems* 15(5) (2007), pp. 560-571.

[70] W. C. Elmore, "The Transient Analysis of Damped Linear Networks with Particular Regard to Wideband Amplifiers", *Journal of Applied Physics* 19(1) (1948), pp. 55-63.

[71] J. Fang and S. S. Sapatnekar, "The Impact of BTI Variations on Timing in Digital Logic Circuits", *IEEE Transactions on Device and Materials Reliability* 13(1) (2012), pp. 277-286.

[72] Z. Feng, P. Li and Z. Ren, "SICE: Design-Dependent Statistical Interconnect Corner Extraction Under Inter/Intra-Die Variations", *IET Circuits, Devices and Systems* 3(5) (2009), pp. 248-258.

[73] D. Fick, N. Liu, Z. Foo, M. Fojtik, J.-S. Seo, D. Sylvester and D. Blaauw, "In Situ Delay-Slack Monitor for High-Performance Processors Using an All-Digital Self-Calibrating $5ps$ Resolution Time-to-Digital Converter", *Proc. International Solid State Circuits Conference*, 2010, pp. 188-189.

[74] F. Firouzi, S. Kiamehr and M. B. Tahoori, "A Linear Programming Approach for Minimum NBTI Vector Selection", *Proc. Great Lakes Symposium on Very Large Scale Integration*, 2011, pp. 253-258.

[75] E. A. Foreman, P. A. Habitz, M.-C. Cheng and C. Visweswariah, "A Novel Method for Reducing Metal Variation with Statistical Static Timing Analysis", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 31(8) (2012), pp. 1293-1297.

[76] J. Franco, B. Kaczer, P. J. Roussel, J. Mitard, S. Sioncke, L. Witters, H. Mertens, T. Grasser and G. Groeseneken, "Understanding the Suppressed Charge Trapping in Relaxed- and Strained-Ge/SiO2/HfO2 PMOSFETs and Implications for the Screening of Alternative High-Mobility Substrate/Dielectric CMOS Gate Stacks", *Proc. IEEE International Electron Devices Meeting*, 2013, pp. 15.2.1-15.2.4.

[77] T. Fukuoka, A. Tsuchiya and H. Onodera, "Worst-Case Delay Analysis Considering the Variability of Transistors and Interconnects", *Proc. ACM International Symposium on Physical Design*, 2007, pp. 35-41.

[78] A. Gattiker, S. Nassif, R. Dinakar and C. Long, "Timing Yield Estimation from Static Timing Analysis", *Proc. International Symposium on Quality Electronic Design*, 2000, pp. 437-452.

[79] R. S. Ghaida and P. Gupta, "Design-Overlay Interactions in Metal Double Patterning", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, Vol. 7275, 2009, pp. 727514-1-727514-10.

[80] J. Ghan, N. Ma, S. Mishra, C. Spanos, K. Poolla, N. Rodriguez and L. Capodieci, "Clustering and Pattern Matching for an Automatic Hotspot Classification and Detection System", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, Vol. 7275, 2009, pp. 727516-1-727516-11.

[81] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming", Version 2.0 beta, 2012. http://cvxr.com/cvx

[82] T. Grasser, B. Kaczer, W. Goes, H. Reisinger, T. Aichinger, P. Hehenberger, P. J. Wagner, F. Schanovsky, J. Franco, M. T. Luque and M. Nelhiebel, "The Paradigm Shift in Understanding the Bias Temperature Instability: From Reaction-Diffusion to Switching Oxide Traps", *IEEE Transactions on Electron Devices* 58(11) (2011), pp. 3652-3666.

[83] P. Gupta, Y. Agarwal, L. Dolecek, N. Dutt, R. K. Gupta, R. Kumar, S. Mitra, A. Nicolau, T. S. Rosing, M. B. Srivastava, S. Swanson and D. Sylvester, "Underdesigned and Opportunistic Computing in Presence of Hardware Variability", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 32(1) (2013), pp. 8-23.

[84] M. Gupta, K. Jeong and A. B. Kahng, "Timing Yield-Aware Color Reassignment and Detailed Placement Perturbation for Bimodal CD Distribution in Double Patterning Lithography", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 29(8) (2010), pp. 1229-1242.

[85] 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 Transactions on Computer-Aided Design of Integrated Circuits and Systems* 26(9) (2007), pp. 1614-1624.

[86] P. Gupta, A. B. Kahng, D. Sylvester and J. Yang, "Performance-Driven Optical Proximity Correction for Mask Cost Reduction", *SPIE Journal of Microlithography, Microfabrication and Microsystems*, 6(3) 2007, pp. 031005-1-031005-8.

[87] S.-J. Han, X. Yu, N. Zamdmer, J. Deng, E. J. Nowak and K. Rim, "Improved Effective Switching Current (I^{+}_{EFF}) and Capacitance Methodology for CMOS Circuit Performance Prediction and Model-to-Hardware Correlation", *Proc. IEEE International Electron Devices Meeting*, 2008, pp. 1-4.

[88] J. Hartmann, "Towards a New Nanoelectronic Cosmology", *Proc. International Solid State Circuits Conference*, 2007, pp. 31-37.

[89] J. Hu, M. C. Fu and S. I. Marcus, "A Model Reference Adaptive Search Method for Global Optimization", *Operations Research* 55(3) (2005), pp. 549-568.

[90] V. Huard, N. Ruiz, F. Cacho and E. Pion, "A Bottom-Up Approach for System-On-Chip Reliability", *Microelectronics Reliability* 51(9-11) (2011), pp. 1425-1439.

[91] F. Huebbers, A. Dasdan and Y. Ismail, "Multi-Layer Interconnect Performance Corners for Variation-Aware Timing Analysis", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2007, pp. 713-718.

[92] K. Jeong and A. B. Kahng, "Timing Analysis and Optimization Implications of Bimodal CD Distribution in Double Patterning Lithography", *Proc. Asia and South Pacific Design Automation Conference*, 2009, pp. 486-491.

[93] K. Jeong and A. B. Kahng, "Methodology From Chaos in IC Implementation", *Proc. International Symposium on Quality Electronic Design*, 2010, pp. 885-892.

[94] K. Jeong, A. B. Kahng, C.-H. Park and H. Yao, "Dose Map and Placement Co-Optimization for Improved Timing Yield and Leakage Power", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 29(7) (2010), pp. 1070-1082.

[95] K. Jeong, A. B. Kahng and K. Samadi, "Quantified Impacts of Guardband Reduction on Design Process Outcomes", *Proc. International Symposium on Quality Electronic Design*, 2008, pp. 790-897.

[96] K. Jeong, A. B. Kahng and R. O. Topaloglu, "Assessing Chip-Level Impact of Double-Patterning Lithography", *Proc. International Symposium on Quality Electronic Design*, 2010, pp. 122-130.

[97] V. Joshi, B. Cline, D. Sylvester, D. Blaauw and K. Agarwal, "Stress Aware Layout Optimization", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 29(5) (2010), pp. 722-736.

[98] H. Jyu and S. Malik, "Statistical Timing Optimization of Combinational Circuits", *Proc. IEEE International Conference on Computer Design*, 1993, pp. 126-137.

[99] C. W. Kaanta, S. G. Bombardier, W. J. Cote, W. R. Hill, G. Kerszykowski, H. S. Landis, D. J. Poindexter, C. W. Pollard, G. H. Ross, J. G. Ryan, S. Wolff and J. E. Cronin, "Dual Damascene: A ULSI Wiring Technology", *Proc. IEEE VLSI Multilevel Interconnection Conference*, 1991, pp. 144-152.

[100] B. Kaczer, S. Mahato, V. V. D. A. Camargo, M. T. Luque, P. J. Roussell, T. Grasser, F. Catthoor, P. Dobrovolny, P. Zuber, G. Wirth and G. Groeseneken, "Atomistic Approach to Variability of Bias-Temperature Instability in Circuit Simulations", *Proc. IEEE International Reliability Physics Symposium*, 2011, pp. XT.3.1-XT.3.5.

[101] A. A. Kagalwalla, P. Gupta, C. J. Progler and S. McDonald, "Design-Aware Mask Inspection", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 31(5) (2012), pp. 690-702.

[102] A. B. Kahng, "Lithography and Design in Partnership: A New Roadmap", *Proc. SPIE/BACUS Symposium on Photomask Technology and Management*, 2008, Vol. 7122, pp. 712202-1712202-10.

[103] A. B. Kahng, "Lithography-Induced Limits to Scaling of Design Quality", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, Vol. 9053, 2014, pp. 905302-1-905302-14.

[104] A. B. Kahng, S. Kang, R. Kumar and J. Sartori, "Enhancing the Efficiency of Energy-Constrained DVFS Designs", *IEEE Transactions on Very Large Scale Integration Systems* 21(10) (2013), pp. 1769-1782.

[105] A. B. Kahng, B. Lin and S. Nath, "Explicit Modeling of Control and Data for Improved NoC Router Estimation", *Proc. IEEE/ACM Design Automation Conference*, 2012, pp. 392-397.

[106] A. B. Kahng, S. Muddu and P. Sharma, "Defocus-Aware Leakage Estimation and Control", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27(2) (2008), pp. 230-240.

[107] A. B. Kahng, C.-H. Park, X. Xu and H. Yao, "Layout Decomposition Approaches for Double Patterning Lithography", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 29(6) (2010), pp. 939-952.

[108] A. B. Kahng, P. Sharma and R. O. Topaloglu, "Chip Optimization Through STI-Stress-Aware Placement Perturbations and Fill Insertion", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27(7) (2008), pp. 1241-1252.

[109] K. Kang, S. P. Park, K. Kim and K. Roy, "On-Chip Variability Sensor Using Phase-Locked Loop for Detecting and Correcting Parametric Timing Failures", *IEEE Transactions on Very Large Scale Integration Systems* 18(2) (2010), pp. 270-280.

[110] M. B. Ketchen, M. Bhushan and D. Pearson, "High Speed Test Structures for In-Line Process Monitoring and Model Calibration", *Proc. International Conference on Microelectronics Test Structures*, 2005, pp. 33-38.

[111] O. Khan and S. Kundu, "A Self-Adaptive System Architecture to Address Transistor Aging", *Proc. Design, Automation and Test in Europe*, 2009, pp. 81-86.

[112] K. K. Kim, W. Wang and K. Choi, "On-Chip Aging Sensor Circuits for Reliable Nanometer MOSFET Digital Circuits", *IEEE Transactions On Circuits and Systems* 57(10) (2010), pp. 798-802.

[113] S. Kim, S. Woo, W. Han, Y. Koh and M. Lee, "Application of Alternating Phase Shift Mask to Device Fabrication", *Proc. SPIE Conference on Optical/Laser Microlithography*, Vol. 2440, 1995, pp. 515-523.

[114] T. I. Kirkpatrick and N. R. Clark, "PERT as an Aid to Logic Design", *IBM Journal of Research and Development* 10(2) 1966, pp. 135-141.

[115] S. Kobayashi and K. Horiuchi, "An LOCV-Based Static Timing Analysis Considering Spatial Correlations of Power Supply Variations", *Proc. Design, Automation and Test in Europe*, 2011, pp. 1-4.

[116] A. Krasnoperova, J. A. Culp, I. Graur, S. Mansfield, M. Al-Imam and H. Maaty, "Process Window OPC for Reduced Process Variability and Enhanced Yield", *Proc. SPIE*, Vol. 6154, 2006, pp. 1200-1211.

[117] K. J. Kuhn, M. D. Giles, D. Becher, P. Kolar, A. Kornfeld, R. Kotlyar, S. T. Ma, A. Maheshwari and S. Mudanai, "Process Technology Variation", *IEEE Transactions on Electron Devices* 58(8) (2011), pp. 2197-2208.

[118] S. H. Kulkarni, D. M. Sylvester and D. T. Blaauw, "Design-Time Optimization of Post-Silicon Tuned Circuits Using Adaptive Body Bias", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27(3) (2008), pp. 481-494.

[119] S. V. Kumar, C. H. Kim and S. S. Sapatnekar, "Adaptive Techniques for Overcoming Performance Degradation Due to Aging in Digital Circuits", *Proc. Asia and South Pacific Design Automation Conference*, 2009, pp. 284-289.

[120] S. V. Kumar, C. H. Kim and S. S. Sapatnekar, "Adaptive Techniques for Overcoming Performance Degradation Due to Aging in CMOS Circuits", *IEEE Transactions on Very Large Scale Integration Systems* 19(4) (2011), pp. 603-614.

[121] R. Kumar and C. P. Ravikumar, "Leakage Power Estimation for Deep Submicron Circuits in an ASIC Design Environment", *Proc. Asia and South Pacific Design Automation Conference*, 2002, pp. 45-50.

[122] T. Kuroda, T. Fujita, S. Mita, T. Nagamatu, S. Yoshioka, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakumu and T. Sakurai, "A $0.9V$ $150MHz$ $10mW$ $4mm^2$ 2-D Discrete Cosine Transform Core Processor with Variable-Threshold-Voltage Scheme", *Proc. International Solid State Circuits Conference*, 1996, pp. 166-168.

[123] A. Kurokawa, H. Masuda, J. Fujii, T. Inoshita, A. Kasebe, Z. Huang and Y. Inoue, "Determination of Interconnect Structural Parameters for Best- and Worst-Case Delays", *IEICE Transactions Fundamentals of Electronics* E89-A(4) (2006), pp. 856-864.

[124] A. Kurokawa, T. Sato, T. Kanamoto and M. Hashimoto, "Interconnect Modeling: A Physical Design Perspective", *IEEE Transactions on Electron Devices* 56(9) (2009), pp. 1840-1851.

[125] L. Lai, V. Chandra, R. Aitken and P. Gupta, "SlackProbe: A Low Overhead In Situ On-Line Timing Slack Monitoring Methodology", *Proc. Design, Automation and Test in Europe*, 2013, pp. 282-287.

[126] S.-C. Lee, A. S. Oates and K. M. Chang, "Limitation of Low-k Reliability Due to Dielectric Breakdown at Vias", *Proc. IEEE International Interconnect Technology Conference*, 2008, pp. 177-179.

[127] S.-S. Lee, E. Boling, A. Kuo and R. Rogenmoser, "A Slew-Rate Based Process Monitor and Bi-Directional Body Bias Circuit for Adaptive Body Biasing in SoC Applications", *Proc. IEEE Custom Integrated Circuits Conference*, 2013, pp. 1-4.

[128] K. T. Lee, W. Kang, E. A. Chung, G. Kim, H. Shim, H. Lee, H. Kim, M. Choe, N. Lee, A. Patel, J. Park and J. Park, "Technology Scaling on High-K & Metal-Gate FinFET BTI Reliability", *Proc. IEEE International Reliability Physics Symposium*, 2013, pp. 2D.1.1-2D.1.4.

[129] Y. Lee and T. Kim, "A Fine-Grained Technique of NBTI-Aware Voltage Scaling and Body Biasing for Standard Cell Based Designs", *Proc. Asia and South Pacific Design Automation Conference*, 2011, pp. 603-608.

[130] S.-C. Lee, A. S. Oates and K.-M. Chang, "Geometric Variability of Nanoscale Interconnects and Its Impact on the Time-Dependent Breakdown of Cu/Low-k Dielectrics", *IEEE Transactions on Device and Materials Reliability* 10(3) (2010), pp. 307-316.

[131] R. Lefferts and C. Jakubiec, "An Integrated Test Chip for the Complete Characterization and Monitoring of a 0.25um CMOS Technology that Fits into Five Scribe Line Structures 150um by 5000um", *Proc. International Conference on Microelectronics Test Structures*, 2003, pp. 59-63.

[132] X. Li, R. Rutenbar and S. Blanton, "Virtual Probe: A Statistically Optimal Framework for Minimum-Cost Silicon Characterization of Nanoscale Integrated Circuits", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2009, pp. 433-440.

[133] L. Liebmann, S. Mansfield, G. Han, J. Culp, J. Hibbeler and R. Tsai, "Reducing DFM to Practice: the Lithography Manufacturability Assessor", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, Vol. 6156, 2006, pp. 61560K-1-61560K-12.

[134] L. Liebmann, B. Grenon, M. Lavin and T. Zell, "Optical Proximity Correction: a First Look at Manufacturability", *Microlithography World* 4(2) (1995), pp. 7-11.

[135] B. Lin, A. Mallik, P. Dinda, G. Memik and R. Dick, "User- and Process-Driven Dynamic Voltage and Frequency Scaling", *Proc. IEEE International Symposium on Performance Analysis of Systems and Software*, 2009, pp. 11-22.

[136] Y. Liu, S. R. Nassif, L. T. Pileggi and A. J. Strojwas, "Impact of Interconnect Variations on the Clock Skew of a Gigahertz Microprocessor", *Proc. IEEE/ACM Design Automation Conference*, 2000, pp. 168-171.

[137] Q. Liu and S. S. Sapatnekar, "A Framework for Scalable Post-Silicon Statistical Delay Prediction Under Spatial Variations", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 28(8) (2009), pp. 1201-1212.

[138] N. Lu and J. McCullen, "Enablement of Variation Aware Timing: Treatment of Parasitic Resistance and Capacitance", *Proc. International Symposium on Quality Electronic Design*, 2007, pp. 743-748.

[139] J. Luo, S. Sinha, Q. Su, J. Kawa and C. Chiang, "An IC-Manufacturing Yield Model Considering Intra-Die Variation", *Proc. IEEE/ACM Design Automation Conference*, 2006, pp. 749-754.

[140] H. Luo, Y. Wang, K. He, R. Luo, H. Yang and Y. Xie, "A Novel Gate-Level NBTI Delay Degradation Model with Stacking Effect", *Proc. International Conference on Integrated Circuit and System Design: Power and Timing Modeling, Optimization and Simulation*, 2007, pp. 160-170.

[141] C. A. Mack, D. A. Legband and S. Jug, "Data Analysis for Photolithography", *Microelectronic Engineering* 46(1-4) (1999), pp. 65-68.

[142] I. A. K. M. Mahfuzul, A. Tsuchiya, K. Kobayashi and H. Onodera, "Variation-Sensitive Monitor Circuits for Estimation of Global Process Parameter Variation", *IEEE Transactions on Semiconductor Manufacturing* 25(4) (2012), pp. 571-580.

[143] S. Mansfield, L. Liebmann, A. Molless and A. K. Wong, "Lithographic Comparison of Assist Feature Design Strategies", *SPIE Optical Microlithography*, Vol. 4000, 2000, pp. 63-76.

[144] S. M. Martin, K. Flautner, T. Mudge and D. Blaauw, "Combined Dynamic Voltage Scaling and Adaptive Body Biasing for Lower Power Microprocessors Under Dynamic Workloads", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2002, pp. 721-725.

[145] D. N. Maynard, D. S. Kerr and C. Whiteside, "Cost of Yield", *Proc. IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop*, 2003, pp. 165-170.

[146] P. McGuinness, "Variations, Margins, and Statistics", *Proc. ACM International Symposium on Physical Design*, 2008, pp. 60-67.

[147] F. D. Meersman, "IC Compiler RM: Reference Methodology with Emphasis on Concurrent MCMM & Signoff Driven Design Closure", *Synopsys Users Group Conference*, San Jose, 2008.

[148] N. Mehta and B. Amrutur, "Dynamic Supply and Threshold Voltage Scaling for CMOS Digital Circuits Using In-Situ Power Monitor", *IEEE Transactions on Very Large Scale Integration Systems*, 2012, pp. 892-901.

[149] M. Meijer, B. Liu, R. V. Veen and J. P. Gyvez, "Post-Silicon Tuning Capabilities of 45nm Low-Power CMOS Digital Circuits", *Proc. VLSI Circuits Symposium Digest of Technical Papers*, 2009, pp. 110-111.

[150] E. Mintarno, J. Skaf, R. Zheng, J. B. Velamala, Y. Cao, S. Boyd, R. W. Dutton and S. Mitra, "Self-Tuning for Maximized Lifetime Energy-Efficiency in the Presence of Circuit Aging", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30(5) (2011), pp. 760-773.

[151] S. Mitra, E. Volkerink, E. J. McCluskey and S. Eichenberger, "Delay Defect Screening Using Monitoring Structures", *Proc. IEEE VLSI Test Symposium*, 2004, pp. 43-48.

[152] A. Mulgaonkar, "Multicorner-Multimode – a Necessary and Manageable Reality of Design", *IC Compiler White Paper*, 2009.
https://www.synopsys.com/apps/protected/docs/pdfs/iccwp/icc_mcmm_wp.pdf

[153] A. Mutlu, J. Le, R. Molina and M. Celik, "Parametric Analysis to Determine Accurate Interconnect Extraction Corners for Design Performance", *Proc. International Symposium on Quality Electronic Design*, 2010, pp. 419-423.

[154] M. H. Na, E. J. Nowak, W. Haensch and J. Cai, "The Effective Drive Current in CMOS Inverters", *Proc. IEEE International Electron Devices Meeting*, 2002, pp. 121-124.

[155] S. R. Nassif, G.-J. Nam and S. Banerjee, "Wire Delay Variability in Nanoscale Technology and Its Impact on Physical Design", *Proc. International Symposium on Quality Electronic Design*, 2013, pp. 591-596.

[156] S. Natarajan, S. Patil and S. Chakravarty, "Path Delay Fault Simulation on Large Industrial Designs", *Proc. IEEE VLSI Test Symposium*, 2006, pp. 1-6.

[157] H. C. Ngo, G. D. Carpenter, A. J. Drake and J. B. Kuang, "Circuit Timing Monitor Having a Selectable-Path Ring Oscillator", *U.S. Patent* No. US7810000B2, October 2010.

[158] L. S. Nielsen, C. Niessen, J. Sparsø and K. V. Berkel, "Low-Power Operation Using Self-Timed Circuits and Adaptive Scaling of the Supply Voltage", *IEEE Transactions on Very Large Scale Integration Systems* 2(4) (1994), pp. 391-397.

[159] J.-P. Noel, O. Thomas, M.-A. Jaud, C. Fenouillet-Beranger, R. Rivallin, P. Scheiblin, T. Poiroux, F. Boeuf, F. Andrieu, O. Weber, O. Faunot and A. Amara, "UT2B-FDSOI Device Architecture Dedicated to Low Power Design Techniques", *Proc. European Solid-State Device Research Conference*, 2010, pp. 210-213.

[160] J. Noguchi, "Dominant Factors in TDDB Degradation of Cu Interconnects", *IEEE Trans. on Electron Devices* 52(8) (2005), pp. 1743-1750.

[161] K. Nose and T. Sakurai, "Analysis and Future Trend of Short-Circuit Power", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 19(9) (2000), pp. 1023-1030.

[162] A. V. Oosten, P. Nikolsky, J. Huckabay, R. Goossens and R. Naber, "Pattern Split Rules! A Feasibility Study of Rule Based Pitch Decomposition for Double Patterning", *Proc. SPIE/BACUS Symposium on Photomask Technology and Management*, Vol. 6730, 2007, pp. 67301-1-67301-7.

[163] M. Orshansky and K. Keutzer, "A General Probabilistic Framework for Worst Case Timing Analysis", *Proc. IEEE/ACM Design Automation Conference*, 2002, pp. 556-561.

[164] K. Ota, M. Saitoh, C. Tanaka and T. Numata, "Threshold Voltage Control by Substrate Bias in 10-nm-Diameter Tri-Gate Nanowire MOSFET on Ultrathin BOX", *Proc. IEEE International Electron Devices Meeting* 34(2) (2013), pp. 187-189.

[165] M. M. Ozdal, C. Amin, A. Ayupov, S. M. Burns, G. R. Wilke and C. Zoo, "An Improved Benchmark Suite for the ISPD-2013 Discrete Cell Sizing Contest", *Proc. ACM International Symposium on Physical Design*, 2013, pp. 168-170.
http://www.ispd.cc/contests/13/ispd2013_contest.html

[166] U. Padmanabhan, J. M. Wang and J. Hu, "Robust Clock Tree Routing in the Presence of Process Variations", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27(8) (2008), pp. 1385-1397.

[167] L.-T. Pang and B. Nikolić, "Measurement and Analysis of Variability in 45nm Strained-Si CMOS Technology", *Proc. IEEE Custom Integrated Circuits Conference*, 2008, pp. 129-132.

[168] C. R. Parthasarathy, M. Denais, V. Huard, G. Ribes, D. Roy, C. Guerin, F. Perrier, E. Vincent and A. Bravaix, "Designing in Reliability in Advanced CMOS Technologies", *Microelectronics Reliability* 46 (2006), pp. 1464-1471.

[169] D. J. Philling and C. Talledo, "In-Situ Monitor of Process and Device Parameters in Integrated Circuits", *U.S. Patent* No. US7583087B2, September 2009.

[170] H. R. Pourshaghaghi and J. P. D. Gyvez, "Power-Performance Optimization using Fuzzy Control of Simultaneous Supply Voltage and Body Biasing Scaling", *Proc. IEEE International Conference on Electronics, Circuits, and Systems*, 2010, pp. 281-284.

[171] K. Qian and C. J. Spanos, "A Comprehensive Model of Process Variability for Statistical Timing Optimization", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, Vol. 6925, 2008, pp. 69251G-1-69251G-11.

[172] K. A. Ramsey, "Tackling the Rising Cost-of-Test for Semiconductor Devices", *Solid State Technology* 54(3) (2011).
http://electroiq.com/blog/2011/03/tackling-the-rising-cost-of-test-for-semiconductor/

[173] R. Rao, A. Devgan, D. Blaauw and D. Sylvester, "Parametric Yield Estimation Considering Leakage Variability", *Proc. IEEE/ACM Design Automation Conference*, 2004, pp. 442-447.

[174] A. Raychowdhury, J. W. Tschanz, K. Bowman, S.-L. Lu, P. Aseron, M. Khellah, B. Geuskens, C. Tokunaga, C. Wilkerson, T. Karnik and V. De, "Error Detection and Correction in Microprocessor Core and Memory Due to Fast Dynamic Voltage Droops", *Journal on Emerging and Selected Topics in Circuits and Systems* 1(3) (2011), pp. 208-217.

[175] S. Reda and S. R. Nassif, "Analyzing the Impact of Process Variations on Parametric Measurements: Novel Models and Applications", *Proc. Design, Automation and Test in Europe*, 2009, pp. 375-380.

[176] D. Reinhard and P. Gupta, "On Comparing Conventional and Electrically Driven OPC Techniques", *Proc. SPIE/BACUS Symposium on Photomask Technology and Management*, Vol. 7488, 2009, pp. 748838-1-748838-8.

[177] B. M. Riess, "Multi-Corner Multi-Mode Synthesis in Design Compiler - A Must or Just Nice to Have?", *Synopsys Users Group Conference*, Germany, 2011.

[178] F. Rigaud, J. M. Portal, H. Aziza, D. Nee, J. Vast, C. Auricchio and B. Borot, "Test Structure for Process and Product Evaluation", *Proc. International Conference on Microelectronics Test Structures*, 2007, pp. 140-144.

[179] W. C. Riordan, R. Miller and E. R. St. Pierre, "Reliability Improvement and Burn In Optimization Through the Use of Die Level Predictive Modeling", *Proc. IEEE International Reliability Physics Symposium*, 2005, pp. 17-21.

[180] A. E. Rosenbluth, S. Bukofsky, C. Fonseca, M. Hibbs, K. Lai, A. F. Molless, R. N. Singh and A. K. K. Wong, "Optimum Mask and Source Patterns to Print a Given Shape", *SPIE Journal of Microlithography, Microfabrication and Microsystems* 1(1) (2002), pp. 13-30.

[181] T. Sakurai and K. Tamaru, "Simple Formulas for Two and Three-Dimensional Capacitances", *IEEE Transactions on Electron Devices* ED-30(2) (1983), pp. 183-185.

[182] S. Saxena, C. Hess, H. Karbasi, A. Rossoni, S. Tonello, P. McNamara, S. Lucherini, S. Minehane, C. Dolainsky and M. Quarantelli, "Variation in Transistor Performance and Leakage in Nanometer-Scale Technologies", *IEEE Transactions on Electron Devices* 55(1) (2008), pp. 131-144.

[183] K. F. Schuegraf and C. Hu, "Oxide Breakdown Model for Very Low Voltages", *IEEE Transactions on Electron Devices* 41(5) (1994), pp. 761-767.

[184] K. Shaik, "Implementation of a Critical Path Based Parametric Ring Oscillator", *BSEE Thesis*, Texas Tech University, Texas, 2011.

[185] A. Sharifi and M. Kandemir, "Process Variation-Aware Routing in NOC Base Multicores", *Proc. IEEE/ACM Design Automation Conference*, 2011, pp. 924-929.

[186] K.-N. Shim, J. Hu and J. Silva-Martinez, "A Dual-Level Adaptive Supply Voltage System for Variation Resilience", *Proc. International Symposium on Quality Electronic Design*, 2010, pp. 38-43.

[187] W. Shiu, W. Ma, H. W. Lee, J. S. Wu, Y. M. Tseng, K. Tsai, C. T. Liao, A. Wang, A. Yau, Y. R. Lin, Y. L. Chen, T. Wang, W. B. Wu and C. L. Shih, "Spacer Double Patterning Technique for Sub-40nm DRAM Manufacturing Process Development", *Proc. SPIE Lithography Asia*, Vol. 7140, 2008, pp. 71403Y-1-71403Y-8.

[188] L. G. Silva, L. M. Silveira and J. R. Phillips, "Efficient Computation of the Worst-Delay Corner", *Proc. Design, Automation and Test in Europe*, 2007, pp. 1617-1622.

[189] J. Srinivasan, S. V. Adve, P. Bose and J. A. Rivers, "The Impact of Technology Scaling on Lifetime Reliability", *Proc. IEEE International Conference on Dependable Systems and Networks*, 2004, pp. 177-186.

[190] L. Stok and J. Cohn, "There is Life Left in ASICs", *Proc. ACM International Symposium on Physical Design*, 2003, pp. 48-50.

[191] M. Stucchi and Z. Tökei, "Impact of LER and Misaligned Vias on the Electric Field in Nanometer-Scale Wires", *Proc. IEEE International Interconnect Technology Conference*, 2008, pp. 174-176.

[192] A. Tetelbaum and S. Chakravarty, "Electronic Design Automation Tool and Method for Optimizing the Placement of Process Monitors in an Integrated Circuit", *U.S. Patent* No. US8010935B2, August 2011.

[193] A. Tiwari and J. Torrellas, "Facelift: Hiding and Slowing Down Aging in Multicores", *Proc. IEEE/ACM Intl. Symposium on Microarchitecture*, 2008, pp. 129-140.

[194] J. W. Tschanz, K. Bowman, S.-L. Lu, P. Aseron, M. Khellah, A. Raychowdhury, B. Geuskens, C. Tokunaga, C. Wilkerson, T. Karnik and V. De, "A $45nm$ Resilient and Adaptive Microprocessor Core for Dynamic Variation Tolerance", *Proc. International Solid State Circuits Conference*, 2010, pp. 282-283.

[195] J. W. Tschanz, J. Kao, S. Narendra, R. Nair, D. Antoniadis, A. Chandrakasan and V. De, "Adaptive Body Bias for Reducing Impacts of Die-to-Die and Within-Die Parameter Variations on Microprocessor Frequency and Leakage", *Proc. International Solid State Circuits Conference*, 2002, pp. 1396-1402.

[196] J. W. Tschanz, S. Narendra, R. Nair and V. De, "Effectiveness of Adaptive Supply Voltage and Body Bias for Reducing Impact of Parameter Variations in Low Power and High Performance Microprocessors", *Journal of Solid State Circuits* 38(5) (2003), pp. 826-829.

[197] R. Vattikonda, W. Wang and Y. Cao, "Modeling and Minimization of PMOS NBTI Effect for Robust Nanometer Design", *Proc. IEEE/ACM Design Automation Conference*, 2006, pp. 1047-1052.

[198] H. J. M. Veendrick, "Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits", *Journal of Solid State Circuits* 19(4) (1984), pp. 468-473.

[199] J. B. Velamala, V. Ravi and Y. Cao, "Failure Diagnosis of Asymmetric Aging Under NBTI", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2011, pp. 428-433.

[200] N. Viswanathan, C. J. Alpert, C. Sze, Z. Li, G.-J. Nam and J. A. Roy, "The ISPD-2011 Routability-Driven Placement Contest and Benchmark Suite", *Proc. ACM International Symposium on Physical Design*, 2011, pp. 141-146.
http://www.ispd.cc/contests/11/ispd2011_contest.html

[201] C. Visweswariah, "Death, Taxes and Failing Chips", *Proc. IEEE/ACM Design Automation Conference*, 2003, pp. 343-347.

[202] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, S. Narayan, D. K. Beece, J. Piaget, N. Venkateswaran and J. G. Hemmett, "First-Order Incremental Block-Based Statistical Timing Analysis", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 25(10) (2006), pp. 2170-2180.

[203] W. Wang, V. Reddy, B. Yang, V. Balakrishnan, S. Krishnan and Y. Cao, "Statistical Prediction of Circuit Aging Under Process Variations", *Proc. IEEE Custom Integrated Circuits Conference*, 2008, pp. 13-16.

[204] X. X. Wang, M. Tehranipoor, S. George, D. Tran, R. Datta and L. Winemberg, "Design and Analysis of a Delay Sensor Applicable to Process/Environmental Variations and Aging Measurements", *IEEE Transactions on Very Large Scale Integration Systems* 20(8) (2012), pp. 1405-1418.

[205] W. Wang, S. Yang and Y. Cao, "Node Criticality Computation for Circuit Timing Analysis and Optimization Under NBTI Effect", *Proc. International Symposium on Quality Electronic Design*, 2009, pp. 763-768.

[206] P. Weckx, B. Kaczer, M. Toledano-Luque, P. Raghavan, J. Franco, P. J. Roussel, G. Groeseneken and F. Catthoor, "Implications of BTI-Induced Time-Dependent Statistics on Yield Estimation of Digital Circuits", *IEEE Transactions on Electron Devices* 61(3) (2014), pp. 666-673.

[207] M.-C. Wu, C.-W. Chiou and H.-M. Hsu, "Scrapping Small Lots in a Low Yield and High-Price Scenario", *IEEE Transactions on Semiconductor Manufacturing* 17(1) (2004), pp. 55-67.

[208] F. Xia, J. He, P. Prabhumirashi, A. Schmitz, A. Lowrie, J. Hicks, Y. Shusterman and R. Brain, "Characterization and Challenge of TDDB Reliability in Cu/Low K Dielectric Interconnect", *Proc. IEEE International Reliability Physics Symposium*, 2011, pp. 2C.1.1-2C.1.4.

[209] L. Xie and A. Davoodi, "Representative Path Selection for Post-Silicon Prediction Under Variability", *Proc. IEEE/ACM Design Automation Conference*, 2010, pp. 593-599.

[210] L. Xie and A. Davoodi, "Bound-Based Statistically-Critical Path Extraction Under Process Variations", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30(1) (2011), pp. 59-71.

[211] K. Yamada and N. Oda, "Statistical Corner Conditions of Interconnect Delay (Corner LPE Specifications)", *Proc. Asia and South Pacific Design Automation Conference*, 2006, pp. 706-711.

[212] E. Yang, C. H. Li, X. H. Kang and E. Guo, "Model-Based Retarget for 45nm Node and Beyond", *Proc. SPIE*, Vol. 7274, 2009, pp. 727428-1-727428-8.

[213] J.-S. Yang, K. Lu, M. Cho, K. Yuan and D. Z. Pan, "A New Graph-Theoretic, Multi-Objective Layout Decomposition Framework for Double Patterning Lithography", *Proc. Asia and South Pacific Design Automation Conference*, 2010, pp. 637-644.

[214] Z. B. Zabinsky and R. L. Smith, "Pure Adaptive Search in Global Optimization", *Mathmatical Programming* 53(1-3) (1992), pp. 323-338.

[215] S. Zafar, Y. H. Kim, V. Narayanan, C. Cabral Jr., V. Paruchuri, B. Doris, J. Stathis, A. Callegari and M. Chudzik, "A Comparative Study of NBTI and PBTI (Charge Trapping) in SiO2/HfO2 Stacks with FUSI, TiN, Re Gates", *Proc. Symposium on VLSI Technology*, 2006, pp. 23-25.

[216] Q. C. Zhang and P. V. Adrichem, "Determining OPC Target Specifications Electrically Instead of Geometrically", *Proc. SPIE*, Vol. 6730, 2007, pp. 67606V-1-67606V-10.

[217] L. Zhang and R. P. Dick, "Scheduled Voltage Scaling for Increasing Lifetime in the Presence of NBTI", *Proc. Asia and South Pacific Design Automation Conference*, 2009, pp. 492-497.

[218] H. Zhang, M. D. F. Wong, K.-Y. K. Chao and L. Deng, "A Practical Low-Power Non-regular Interconnect Design with Manufacturing for Design Approach", *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 2(2) (2012), pp. 322-332.

[219] W. Zhao and Y. Cao, "New Generation of Predictive Technology Model for Sub-45nm Design Exploration", *Proc. International Symposium on Quality Electronic Design*, 2006, pp. 585-590.

[220] W. Zhao, Y. Cao, F. Liu, K. Agarwal, D. Acharyya, S. Nassif and K. Nowka, "Rigorous Extraction of Process Variations for 65nm CMOS Design", *Proc. IEEE Conference on Solid State Device Research*, 2007, pp. 89-92.

[221] C. Zhuo, D. Blaauw and D. Sylvester, "Variation-Aware Gate Sizing and Clustering for Post-Silicon Optimized Circuits", *Proc. International Symposium on Low Power Electronic Design*, 2008, pp. 105-110.

[222] V. Zolotov, J. Xiong, H. Fatemi and C. Visweswariah, "Statistical Path Selection for At-Speed Test", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2008, pp. 624-631.

[223] P. S. Zuchowski, P. A. Jabitz, J. D. Hayes and J. J. Oppold, "Process and Environmental Variation Impacts on ASIC Timing", *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2004, pp. 336-342.

[224] *ARM Cortex-M0 processor*,
http://www.arm.com/products/processors/cortex-m/cortex-m0.php

[225] *ARM Cortex-M3 processor*,
http://www.arm.com/products/processors/cortex-m/cortex-m3.php

[226] *Cadence SoC Encounter User Guide*,
http://www.cadence.com/products/di/firstencounter/pages/default.aspx

[227] *Cadence ICT*, http://www.cadence.com/Community/tags/ICT/default.aspx

[228] *Mentor Calibre*, https://www.mentor.com

[229] *ICknowledge*, http://www.icknowledge.com

[230] *ISCAS-85 Benchmark Circuits Verilog Files*,
http://www.pld.ttu.ee/ maksim/benchmarks/iscas85/verilog

[231] *International Technology Roadmap for Semiconductors*, 2007 Edition,
http://public.itrs.net

[232] *International Technology Roadmap for Semiconductors*, 2009 Edition,
Interconnect Chapter, Table INTC6,
http://www.itrs.net/Links/2009ITRS/2009Chapters_2009Tables/2009_Interconnect.pdf

[233] *International Technology Roadmap for Semiconductors*, 2011 Edition, System Driver Chapter, http://www.itrs.net/Links/2011ITRS/2011Chapters_2011Chapters/2011_SysDrivers.pdf

[234] *International Technology Roadmap for Semiconductors*, 2011 Edition, Interconnect Chapter, Table INTC6, http://www.itrs.net/Links/2011ITRS/2011Tables/Interconnect_2011Tables.xlsx

[235] *International Technology Roadmap for Semiconductors*, 2011 Edition, Process Integration, Devices, and Structures Chapter, Table PIDS2, http://www.itrs.net/Links/2011ITRS/2011Tables/PIDS_2011Tables.xlsx

[236] *JEDEC*, "Temperature, Bias and Operating Life", JESD22-A108D, 2010, http://www.jedec.org

[237] *Liberty Technical Advisory Board*, http://www.si2.org/Liberty-TAB

[238] *lp_solve reference guide*, http://lpsolve.sourceforge.net/5.5

[239] *MATLAB*, http://www.mathworks.com/products/matlab

[240] *Nangate Open Cell Library*, https://www.nangate.com

[241] *Openaccess API*, http://www.si2.org

[242] *OpenSPARC T1 Project*, http://www.sun.com/processors/opensparc

[243] *OpenCores*, http://opencores.org

[244] *Sensitivity-Based Leakage Optimizer*, http://vlsicad.ucsd.edu/SIZING/optimizer.html#SensOpt

[245] *SPICE*, http://bwrcs.eecs.berkeley.edu/Classes/IcBook/SPICE/

[246] *Stanford CPUDB*, http://cpudb.stanford.edu

[247] *Sun OpenSPARC T1 Project*, http://www.sun.com/processors/opensparc

[248] *Synopsys 32/28nm Generic Library*, http://www.synopsys.com/COMMUNITY/UNIVERSITYPROGRAM/Pages/32-28nm-generic-library.aspx

[249] *Synopsys Advanced OCV Technology*, http://www.synopsys.com/Tools/Implementation/SignOff/CapsuleModule/PrimeTime_AdvancedOCV_WP.pdf

[250] *Synopsys Design Compiler Users Manual*, http://www.synopsys.com

[251] *Synopsys HSPICE Users Manual*, http://www.synopsys.com

[252] *Synopsys IC Compiler*, http://www.synopsys.com

[253] *Synopsys Interconnect Technology Format*, http://www.synopsys.com/community/interoperability/pages/tapinitf.aspx

[254] *Synopsys PrimeTime*,
http://www.synopsys.com/Tools/Implementation/SignOff/PrimeTime/Pages/default.aspx

[255] *Synopsys Raphael*,
http://www.synopsys.com/Tools/TCAD/InterconnectSimulation/Pages/Raphael.aspx

[256] *Synopsys SiliconSmart*, http://www.synopsys.com

[257] *Texas Instruments PowerWise*,
http://www.ti.com/ww/en/analog/power_management/powerwise-avs.shtml.