UC San Diego
UC San Diego Electronic Theses and Dissertations

Title
On compensation of systematic manufacturing variations in physical design

Permalink
https://escholarship.org/uc/item/9tr5k1cb

Author
Gupta, Puneet

Publication Date
2007

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA, SAN DIEGO

On Compensation of Systematic Manufacturing Variations in Physical Design

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

by

Puneet Gupta

Committee in charge:

Professor Andrew B. Kahng, Chair
Professor Chung-Kuan Cheng
Professor Sujit Dey
Professor Rajesh Gupta
Dr. Fook-Luen Heng
Professor Dennis Sylvester
Professor Yuan Taur

2007

Copyright

Puneet Gupta, 2007

All rights reserved.

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

____

____

____

____

____

____

____

Chair

University of California, San Diego

2007

DEDICATIONS

This dissertation is dedicated to my mentors, teachers, friends and family, whose constant motivation and guidance have helped me throughout my research.

To Professor Andrew Kahng, my advisor, the best I could have ever hoped for. For the knowledge and perspective he provided. For always pushing me. For always being interested in my ideas, howsoever bad, and helping me get them to fruition if they made sense.

To Professor Dennis Sylvester for his constant guidance on many of my projects and for helping inculcate device and circuit intuition in me. For always being willing to be a sounding board for ideas.

To Dr. Fook-Luen Heng and Dr. Daniel Ostapko for giving me the amazing learning opportunity of working in one of the most reputed research centers in the world. To Dr. Fook-Luen Heng, Dr. Ruchir Puri, Dr. Mark Lavin and other researchers at IBM for several enlightening discussions.

To all my collaborators for all the work that we did together. To Saumil, Youngmin, Swamy, Puneet, Jie and Dr. Ion Mandoiu for all their interesting ideas and everything I learned from them.

To my labmates for the countless hours we spent together in AP&M. To Swamy, Puneet and Sherief for the weekend Indian buffet lunches and midnight dinners at Cotixan.

To my friends and roommates for their company and friendship. To Satya for always being ready to take a coffee trip. To Rakesh for the numerous engaging conversations including but not limited to research.

To my colleagues at Blaze DFM for making my time there a big learning experience and opportunity for industrial application of my research.

To my thesis committee members for providing valuable feedback on my research.

Finally, to my family for their constant love, support and encouragement throughout my life even when they were thousands of miles away.

TABLE OF CONTENTS

Signature Page . . . iii

Dedications . . . iv

Table of Contents . . . v

List of Figures . . . viii

List of Tables . . . xii

Acknowledgments . . . xiv

Vita and Publications . . . xvii

Abstract of the Dissertation . . . xxi

I Introduction . . . 1
- A. Taxonomy of Yield Loss . . . 1
- B. Sources of Yield Loss . . . 3
- C. Common Methods for Yield Optimization . . . 5
  - 1. Design Rules . . . 5
  - 2. Corner-Based Design Analysis . . . 6
- D. Systematic Variations . . . 8
  - 1. Chemical Mechanical Polishing (CMP) Effects . . . 8
  - 2. Photolithography Effects . . . 9
- E. This Thesis . . . 12
- F. Acknowledgments . . . 14

II Dealing with FEOL Forbidden Pitches in Detailed Placement . . . 15
- A. Assist Feature Correctness . . . 16
  - 1. RET and Layout Impact . . . 16
  - 2. SRAF Rule and Forbidden Pitch Extraction . . . 19
  - 3. AFCorr Placement Algorithm . . . 20
  - 4. Experimental Setup and Results . . . 25
- B. Etch Dummy Correctness . . . 31
  - 1. Etch Dummy and Layout Impact . . . 31
  - 2. SRAF-Aware Etch Dummy Generation . . . 34
  - 3. Corr Placement Algorithm . . . 35
- C. Modified Design and Evaluation Flow . . . 38
- D. Conclusions and Ongoing Work . . . 43
- E. Acknowledgments . . . 46

III Dealing with Systematic Focus-Dependent FEOL CD Variation 47
A. Systematic Variation Aware Timing Analysis 48
1. Systematic Variation: Magnitude and Impact 49
2. Overview of the Systematic Variation Aware Static Timing Methodology 51
3. Experiments and Results 58
4. Practical Systematic Variation Aware Timing 60
5. Conclusions and Ongoing Work 61
B. Self-Compensating Design for Focus Variation 62
1. Layout Generation 63
2. Self-Compensated Design Baselines 69
3. Optimization (Self-Compensated Physical Design) 70
4. Results 75
5. Conclusions 79
C. Acknowledgments 80

IV Dealing With FEOL Leakage Variability by Gate-Length Biasing 84
A. Cell-Level Gate-Length Biasing 87
1. Library Generation 87
2. Optimization for Leakage 89
B. Experiments and Results 92
1. Leakage Reduction 93
2. Manufacturability and Process Effects 97
3. Process Variability 100
C. Conclusions and Ongoing Work 102
D. Standard-Cell Library Optimization for Leakage Reduction 103
1. Biasing Objectives and Cell-variants 105
2. Variant List Pruning 108
3. Biasing Methodology 109
4. Delay/Leakage Models 113
5. Optimization setup and Results 114
6. Conclusions 116
E. Acknowledgments 117

V The Loop Back from Lithography Simulation 118
A. Lithography Simulation-Based Full-Chip Design Analyses 118
1. Device Analyses 120
2. Interconnect Analyses 125
3. Full-Chip Analyses 126
4. Experiments and Results 128
5. Conclusions 129
B. Modeling of Non-Uniform Device Geometries for Post-Lithography Circuit Analysis 132
1. Physical Explanation of the Edge Effect 133
2. Development of Location-Dependent V_{th} Model 133
3. TCAD Setup for Model Accuracy Verification 137
4. Results 141

5. Conclusions . . . 143
C. Acknowledgments . . . 145

VI The Cost Angle . . . 146
A. Performance-Driven OPC for Mask Cost Reduction . . . 146
1. General Cost of Correction Flow (MinCorr) Based on Sizing . . . 149
2. Experimental Setup and Results . . . 157
3. Conclusions and Future Work . . . 162
B. Modeling OPC Complexity for Design for Manufacturability . . . 163
1. Library MCC (LMCC) . . . 164
2. Wire MCC (WMCC) . . . 169
3. Conclusions . . . 178
C. Acknowledgments . . . 178

VII Coping with BEOL Variability: Performance-Aware Metal Fill Synthesis 179
A. Related Work . . . 181
B. Capacitance and Delay Models . . . 183
C. Problem Formulations . . . 186
1. Min-Delay-Fill-Constrained Objective . . . 186
2. Max-MinSlack-Fill-Constrained Objective . . . 187
D. Geometry Computation . . . 188
E. Approaches for MDFC PIL-Fill . . . 189
1. Integer Linear Programming Approach . . . 189
2. Greedy Method . . . 192
F. Iterated Approach for MSFC PIL-Fill . . . 192
G. Computational Experience . . . 194
1. MDFC PIL-Fill Experiments . . . 194
2. MSFC PIL-Fill Experiments . . . 195
H. Conclusions and Future Research . . . 196
I. Acknowledgments . . . 196

Bibliography . . . 199

LIST OF FIGURES

Figure I.1: The CMP process. 9
Figure I.2: Impact of metal fill on topography. 9
Figure I.3: A picture showing OPC and alternating PSM. Also notice the SRAFs next to the OPC'd feature. 11

Figure II.1: Comparison of Bossung plots between dense and isolated lines: (a) results of Bias OPC and (b) results of SRAF OPC. . . . 17
Figure II.2: (a) H-interactions of gate-to-gate, gate-to-field, and field-to-gate, (b) overlapped area of (a), (c) V-interactions of field-to-field-polys. 21
Figure II.3: Horizontal Cost ($HCost$) calculation. 22
Figure II.4: Vertical Cost ($VCost$) calculation. 23
Figure II.5: Cell placement before and after horizontal AFCorr 25
Figure II.6: Cell placement before and after vertical AFCorr 26
Figure II.7: Through-pitch proximity plots for 130 nm technology: best focus without OPC, worst defocus without OPC, worst defocus with BIAS OPC, and worst defocus with SRAF OPC are shown. 28
Figure II.8: Through-pitch proximity plots and etch skew for 90 nm technology: worst defocus with SRAF OPC and worst defocus with etch OPC (left Y-axis), and etch bias (right Y-axis) are shown. 28
Figure II.9: Number of SRAFs with and without AFCorr for each of five different utilizations. 31
Figure II.10: Reductions of forbidden pitches with AFCorr methodology for each of five different utilizations. 31
Figure II.11: Number of inserted SRAF and etch dummy features with various etch dummy insertion methodologies for each of five different utilizations. 32
Figure II.12: Different proximity behaviors between photo and etching processes with pitch. 33
Figure II.13: Conflict between SRAF and etch dummy rules: (a) assist feature missing, (b) forbidden pitch occurrence. 33
Figure II.14: (a) Typical etch dummy generation, (b) SRAF-aware etch dummy generation. 34
Figure II.15: The placement perturbation problem for assist and etch dummy insertion. (a) Multiple interactions of gate-to-dummy and field-to-dummy, (b) Overlap area when there is no etch dummy, (c) Overlap area in the presence of etch dummy. 37
Figure II.16: The algorithm for $AFCost$ and $EDCost$ computations. . . 39
Figure II.17: The modified design and evaluation flows: Note the added steps of forbidden pitch extraction, SAEDM and post-placement optimization to ASIC design flow. 40
Figure II.18: Calibrated vertical profile after photo and etch processes. . 41
Figure II.19: Reductions of forbidden pitches with various etch dummy insertion methodologies for each of five different utilizations. . . . 42

Figure III.1: An example of through-pitch variation for an annular illumination system with λ=193nm and NA=0.7 calculated using *Prolith* [57]. The drawn dimension is 130 nm. Notice the "radius of influence" of less than 600nm. 49
Figure III.2: Linewidth vs. defocus for 0.35 μm width with varying spacing for a given exposure setting [55]. Notice the systematically different behavior or isolated and nested lines. 50
Figure III.3: A library-based OPC environment setup for a simple NAND gate. Note the dummy poly geometries inserted to emulate the impact of neighboring cells on the cell under consideration. 53
Figure III.4: An example placement of cells A, B and C. For cell B, $nps_B^{LT} = 900, nps_B^{RT} = 950, nps_B^{LB} = 750, nps_B^{RB} = 900$. 53
Figure III.5: Mixture of dense, isolated and self-compensated devices. . . 55
Figure III.6: An artificial Bossung curve at some given nominal exposure. The smile denotes the "most dense" feature in the technology while the frown denotes the "most isolated" one. It should be clear that the total span of CD variation ($= 2(lvar_{pitch} + lvar_{focus})$) is too pessimistic. 57
Figure III.7: Distribution of error for model-based OPC for c3540 ISCAS85 benchmark. 59
Figure III.8: Linewidth variation with spacing (SBs are inserted at 420nm and 660nm). 63
Figure III.9: Linewidth variation with asymmetric spacing for two defocus values, 0.0um and 0.4um. The nearly flat surface represents 0.0um defocus. 64
Figure III.10: Linewidth variation with defocus level (nominal linewidth = 130 nm). 66
Figure III.11: Linewidth with spacing from 0.5μm to 2μm at 0.0μm and 0.4μm defocus in Case 2 for dense and iso cells. 68
Figure III.12: Linewidth variation at 0.4μm defocus in Case 1. Arrows indicate scattering bar (SB) insertion points. 69
Figure III.13: Illustration of optimization process (D denotes "dense" and I denotes "iso" cell; numbers are example sensitivities of gates to swapping to "iso" counterparts). 71
Figure III.14: Optimization algorithm for self-compensated design. 71
Figure III.15: Slack vs. defocus for benchmark c7552 showing the effectiveness of various self-compensating design options. Note some defocus values (e.g., 0.1-0.18μm) at which the circuit fails to meet timing requirement under the heuristic optimization without post-processing. The horizontal line at y=0 is added to highlight the timing constraint. 78
Figure III.16: Stacked histograms showing the delay distribution for c6288 (required time = 4.68ns). Note that there is a break in the y-axis at 21. 79

Figure IV.1: Variation of leakage and delay (each normalized to 1.00) for an NMOS device in an industrial 130nm technology. 85
Figure IV.2: Pseudocode for cell-level gate-length biasing for leakage optimization. 91
Figure IV.3: Cell layout of a generic AND2X6 with simulated printed gate-lengths. 99
Figure IV.4: Leakage distributions for unbiased, uniform-biased and technology-level selectively-biased alu128. Note the "left-shift" of the distribution with the introduction of biased devices in the design. 102
Figure IV.5: I_{off}/I_{on} characteristics for PMOS and NMOS devices. . . . 104
Figure IV.6: AND circuit diagram. 109
Figure IV.7: Basic biasing algorithm. 111
Figure IV.8: Pre and post optimization leakage distribution for AES. . . 115

Figure V.1: Three possible ways for rectilinearization. 122
Figure V.2: Steps involved in shape simplification for capacitance computation. 126
Figure V.3: Lithography simulation-based design analyses flow. 127
Figure V.4: Cross-section of device showing STI edges and fringing capacitances. 134
Figure V.5: Comparison of I_{off} vs. width characteristics of observed data and fitted model. 137
Figure V.6: Comparison of I_{on} vs. width characteristics of observed data and fitted model. 138
Figure V.7: V_{th} as a function of location. 139
Figure V.8: On and Off current densities as a function of position along channel. 140
Figure V.9: I_{on} and I_{off} as a function of width for SPICE and Davinci setups. 140
Figure V.10: Davinci structure used for verification of results diagrammatic representation and 3D Device Structure. 141
Figure V.11: I_{off} contours vs. protrusion dimensions. 144

Figure VI.1: Relative contributions of various components of mask cost [157]. 151
Figure VI.2: The signed edge placement error (EPE). 151
Figure VI.3: An example of three levels of OPC [158]. (a) No OPC, (b) Medium OPC, (c) Aggressive OPC. 152
Figure VI.4: Mask data volume (kB) vs. EPE tolerance for a NAND3X4 cell in TSMC 130 nm technology. 152
Figure VI.5: The EPEMinCorr flow to find quantified edge placement error tolerances for layout features and drive OPC with them. 153
Figure VI.6: Comparison of average printed gate CD with and without pre-bias for the cell macro NAND3X4. 156
Figure VI.7: Summary of EPE assigment for OPC level control. 161

Figure VI.8: Gate CD distribution for c432. Gates with budgeted 4 nm EPE tolerance are labeled critical gates while others are labeled as non-critical. The y-axis shows the number of fragments of gate edges with a given printed CD. 161
Figure VI.9: Fracture count variation with IT and OT for maximum fragment edgelengths of 50nm, 100 nm and 200nm. 166
Figure VI.10: Flow chart of LMCC methodology. 167
Figure VI.11: Poly and active regions of two standard cells. 168
Figure VI.12: Pair-wise scatter plot showing trends between poly fracture count (PFC) and layout parameters and between different layout parameters. 169
Figure VI.13: Scatter plot showing actual PFC (dots) versus fit (line) based on three variables, NP, PVC and PW. 170
Figure VI.14: Predicted vs. actual FC of 100 standard cells. The prediction is based on model built using 15 standard cells only. 170
Figure VI.15: An example of context with line-body, line-end and corner shapes. 171
Figure VI.16: An example of context for WMC glossary. 171
Figure VI.17: Test structures for (a) line-end characteristic length (LECL) and (b) corner characteristic length (CCL). 173
Figure VI.18: General trend in FC saturation with (a) S, (b) NS, (c) NL, (d) LECL and (e) CCL. 173
Figure VI.19: Two examples for validation of line-body model: (a) two neighbors with same space ($S1 = 200nm$) and (b) two neighbors with different spaces of ($S1 = 200nm$ and $S2 = 400nm$). 174
Figure VI.20: Line-body model: (a) Test patterns for α calculation, (b) test patterns for β LUTs generation and (c) line-body model ($FC = \alpha\text{ML} + \beta(S, EPE_{tol})$). 175
Figure VI.21: Pattern examples for line-end model: (a) line-end with single neighbor and (b) line-end with double neighbors. 176
Figure VI.22: Three examples for line-end model validation: (a) line-end with S = 200nm, (b) line-end perpendicular neighbor, (c) line-end with S1 = 200nm and S2 = 400nm. 176

Figure VII.1: In the fixed r-dissection framework, the n-by-n layout is partitioned by r^2 (here, $r = 3$) distinct overlapping dissections with window size $w \times w$. This induces $\frac{nr}{w} \times \frac{nr}{w}$ *tiles*. Each dark-bordered $w \times w$ window consists of r^2 tiles. 180
Figure VII.2: Example configurations of floating dummy fill. 184
Figure VII.3: Segmented RC line model. 185
Figure VII.4: SlackColumn-III definition: Illustration of scan-line and slack blocks within tile between pairs of active lines in adjacent tiles. . 188
Figure VII.5: Scan-line algorithm to find fill slack columns on given layer (assuming horizontal routing direction). 189
Figure VII.6: Greedy MDFC PIL-Fill algorithm. 197
Figure VII.7: Greedy MSFC PIL-Fill algorithm. 198

LIST OF TABLES

Table II.1: SRAF rule table in $0.13\mu m$ and $0.09\mu m$ lithography. 28
Table II.2: Summary of Forbidden pitch results. Forbidden pitch counts slightly change based on different H- vs. V-weights. 29
Table II.3: Summary of AFCorr results. Runtime denotes the runtime of SRAF and etch dummy insertion and model-based OPC. The AFCorr perturbation runtime ranges from 2 to 3 minutes for all test cases. GDS size is the post-SRAF OPC data volume. 30
Table II.4: Comparison of etch dummy rules between conventional etch dummy method and SAEDM. Note that AS_l + AS_r = ES - ED_l. 36
Table II.5: Etch process conditions for the simulator in 90 nm technique. 41
Table II.6: Forbidden pitch results with various etch dummy insertion methodologies in resist and etch processes. 43
Table II.7: Summary of SAEDM+Corr results. Runtime denotes the runtime of SRAF and etch dummy insertion, as well as model-based OPC. The Corr perturbation runtime ranges from 4 to 5 minutes for all test cases. GDS size is the post-OPC data volume. 44

Table III.1: Comparison of library-based OPC and full-chip OPC. N-i% denotes % of devices with less than i% error compared to full-chip OPC. library OPC Runtime is 90 seconds for 10 masters. 59
Table III.2: Comparison of traditional worst-case timing with systematic variation aware timing methodology. Nom, BC, WC denote nominal, best-case and worst-case corners of the library respectively. . . 60
Table III.3: Parameters used in CalibreWB. 64
Table III.4: Spacing criteria for cell generation (SB = Scattering Bar). . 66
Table III.5: Normalized area overhead of each cell version. 68
Table III.6: Area and leakage power change when dense cells are exchanged with iso Counterparts. 73
Table III.7: Top 5 most swapped gates for circuit c5315 by each approach. 77
Table III.8: Normalized delay and leakage power for ISCAS85 benchmark circuits synthesized in each library type (normalized to original cells at 0.0μm defocus value). 81
Table III.9: Normalized area and gate distribution for each library and optimization approach. 82
Table III.10: Leakage power change for self-compensating designs and two heuristic-based optimizations at 0.4μm defocus compared to the original library at 0.0μm defocus. 83

Table IV.1: Comparison of leakage and runtime when EISTA and CISTA are used for sensitivity computation. 92
Table IV.2: Test cases used in our experiments and their details. 92
Table IV.3: Leakage reduction and delay penalty due to gate-length biasing for all 25 cells in our library. 94

Table IV.4: Impact of gate-length biasing on leakage and dynamic power (assuming an activity factor of 0.02) for single threshold-voltage designs. Delay penalty constraint is set to 0%, 2.5%, and 5% for each of the test cases. (Note: delay penalty for SVT-SGL is always set to 0% due to the non-availability of V_{th} and $Lgate$ knobs. SVT-DGL is slower than SVT-SGL for delay penalties of 2.5% and 5%.) 95
Table IV.5: Impact of gate-length biasing on leakage and dynamic power (assuming an activity factor of 0.02) for dual threshold-voltage designs. Delay penalty constraint is set to 0%, 2.5%, and 5% for each of the test cases. 96
Table IV.6: Impact of gate-length biasing on subthreshold leakage and gate tunneling leakage of $90nm$ PMOS and NMOS devices of $1\mu m$ width at different temperatures. Total leakage reductions are high even when gate leakage is considered. 98
Table IV.7: Comparison of printed dimensions of unbiased and biased versions of AND2X6. The unbiased nominal gate-length is $130nm$ while the biased nominal is $138nm$. Note the high correlation between unbiased and biased versions. 100
Table IV.8: Process window improvement with gate-length biasing. The CD tolerance is kept at $13nm$. ELAT=Exposure latitude. 100
Table IV.9: Reduction in performance and leakage power uncertainty with biased gate-length in the presence of inter-die variations. The uncertainty spread is specified as a percentage of nominal. The results are given for dual V_{th} and the biasing is $8nm$. 101
Table IV.10: Slack characteristics of circuit timing reports. All values are in ps. 107
Table IV.11: List of variants and polarity of biases. 107
Table IV.12: Distribution of states over different devices in AND gate. . . 107
Table IV.13: Average delay and leakage overheads for all variants. 112
Table IV.14: Optimization results for TLB and CLB based libraries. . . . 115

Table V.1: Transistor-level modeling matching accuracy. 125
Table V.2: Testcases used in our experiments. 128
Table V.3: Delay, leakage, and dynamic power estimates after layout, and after lithography simulation at $0nm$ and $100nm$ defocus using the proposed flow. 130
Table V.4: Accuracy of mixed-mode analyses with respect to individual objective-specific analyses. Circuit c5315 is combinational so hold-time analysis is not applicable. 131
Table V.5: TCAD model parameters. 138
Table V.6: Results for non-rectilinear gates with protrusion or depression of width 20nm. 142
Table V.7: Results for non-rectilinear gates with protrusion or depression of width 40nm. 142

Table VI.1: The ITRS requirement of gate dimension variation control is becoming more stringent as the technology scales. 148

Table VI.2: Correspondence between the traditional gate sizing problem and the minimum cost of correction (to achieve a prescribed selling point delay with given yield) problem. 149
Table VI.3: Benchmark details. 157
Table VI.4: Impact of EPEMinCorr optimization on cost and CD. All runtimes are based on a 2.4GHz Xeon machine with 2GB memory running Linux. 160
Table VI.5: Optical model parameters. 168
Table VI.6: OPC parameters. 168
Table VI.7: WMC glossary. 171
Table VI.8: Saturation point characteristics: $S_0 = 0.65\mu$m, $NS_0 = 0.13\mu$m, $NL_0 = 0.13\mu$m, $LECL_0 = 0.65\mu$m and $CCL_0 = 0.65\mu$m. 173
Table VI.9: LUT for line-body model and comparison of FC with simulation and experimental results for two examples: maximum difference of FC is 5. 175
Table VI.10: LUTs for single and double neighbors cases of the line-end model: $LECL_0$ is 650nm. 177
Table VI.11: Comparison of predicted FC with real FC for the three line-end test patterns in Figure VI.22. 177
Table VI.12: Real versus predicted FC for different EPE_{tol} of metal layer 2 from ALU128 benchmark implemented in the 90 nm technology. 177

Table VII.1: Weighted MDFC PIL-Fill synthesis. **Notation:** *T/W/r:* testcase / window size / r dissection; *Normal:* normal fill result; *ILP:* Look-up Table Based Integer Linear Programming method; *Greedy:* Greedy method; τ*:* total delay increase (ns); *CPU:* runtime of PIL-Fill step (seconds). 195
Table VII.2: Iterated approaches for MSFC PIL-Fill. **Notation:** *MaxDen:* maximum window density on layout; *MinDen:* minimum window density on layout; *DenConstr:* density requirement specified as a minimum post-fill window density; *MSFC-PIL:* results of MSFC PIL-Fill method; *minSlack:* minimum slack over all nets (ps). . . 195

ACKNOWLEDGMENTS

Chapter I is in part a reprint of "Yield Analysis and Optimization", *The Handbook of Algorithms for VLSI Physical Design Automation*, CRC Press, to be published. I would like to thank my coauthor Dr. Evanthia Papadopoulou.

Chapter II is in part a reprint of "Detailed Placement for Enhanced Control of Resist and Etch CDs", to appear in *IEEE Transactions on CAD*. I would like to thank my coauthors Chul-Hong Park and Dr. Andrew B. Kahng.

Chapter III is in part reprint of "Self-Compensating Design for Reduction of Timing and Leakage Sensitivity to Systematic pattern-dependent Variation", to appear in *IEEE Transactions on CAD* and "Toward a Systematic-Variation Aware Timing Methodology", *Proc. ACM/IEEE Design Automation Conference*, 2004. I would like to thank my coauthors Youngmin Kim, Dr. Dennis Sylvester, Dr. Fook-Luen Heng and Dr. Andrew B. Kahng.

Chapter IV is in part a reprint of "Gate-Length Biasing for Runtime Leakage Control", *IEEE Transactions on Computer-Aided Design*, 25(8) (2006) and "Standard-Cell Library Optimization for Leakage Reduction", *Proc. ACM/IEEE Design Automation Conference*, 2006. I would like to thank my coauthors Puneet Sharma, Saumil Shah, Dr. Dennis Sylvester and Dr. Andrew B. Kahng.

Chapter V is in part a reprint of "Lithography Simulation-Based Full-Chip Design Analyses", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2006 and "Modeling of Non-Uniform Device Geometries for Post-Lithography Circuit Analysis", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2006. I would like to thank my coauthors Puneet Sharma, Saumil Shah, Youngmin Kim, Dr. O. Sam Nakagawa, Dr. Dennis Sylvester and Dr. Andrew B. Kahng.

Chapter VI is in part a reprint of "Modeling OPC Complexity for Design for Manufacturability", *Proc. BACUS Symposium on Photomask Technology and Management*, 2005 and "Performance-Driven OPC for Mask Cost Reduction", submitted to *SPIE Journal of Microlithography, Microfabrication and Microsystems*. I would like to thank my coauthors Chul-Hong Park, Swamy Muddu, Dr. O. Sam Nakagawa, Jie Yang, Dr. Dennis Sylvester and Dr. Andrew B. Kahng.

Chapter VII is in part a reprint of "Performance-Impact Limited Area Fill Synthesis", *Proc. ACM/IEEE Design Automation Conference*, 2003. I would like to thank my coauthors Dr. Yu Chen and Dr. Andrew B. Kahng.

VITA

1979	Born, Alwar, Rajasthan (INDIA).
1996	High School Certificate, Jaipur, Rajasthan (INDIA).
Summer 1999	Internship, Texas Instruments, Bangalore (INDIA).
2000	B.Tech in Electrical Engineering, Indian Institute of Technology (IIT), New Delhi.
2000–2001	VLSI Design Engineer, Mindtree Technologies, Bangalore (INDIA).
Summer 2003	Internship, IBM T.J. Watson Research Center, Yorktown Heights, New York.
2004	M.S., University of California, San Diego.
Summer 2004	Internship, IBM T.J. Watson Research Center, Yorktown Heights, New York.
October 2004 - Present	Product Architect, Blaze DFM, Inc., Sunnyvale, California.
2007	Ph.D., University of California, San Diego

PUBLICATIONS

P. Gupta, A. B. Kahng and S. Mantik, “Routing-Aware Scan Chain Ordering”, *ACM Transactions on Design Automation of Electronic Systems*, 10(3) (2005).

P. Gupta, A. B. Kahng, I. I. Mandoiu and P. Sharma, “Layout-Aware Scan Chain Synthesis for Improved Path Delay Fault Coverage”, *IEEE Transactions on Computer-Aided Design*, 24(7) (2005).

P. Gupta, A. B. Kahng and S. Muddu, “Quantifying Error in Dynamic Power Estimation of CMOS Circuits”, *Journal of Analog Integrated Circuits and Signal Processing*, 42(3) (2005).

P. Gupta, A. B. Kahng, P. Sharma and D. Sylvester, “Gate-Length Biasing for Runtime Leakage Control”, *IEEE Transactions on Computer-Aided Design*, 25(8) (2006).

P. Gupta, A. B. Kahng, C.-H. Park, K. Samadi and X. Xu, “Wafer Topography-Aware Optical Proximity Correction”, to appear in *IEEE Transactions on Computer-Aided Design*.

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", to appear in IEEE Transactions on CAD.

P. Gupta, A. B. Kahng and C.-H. Park, "Detailed Placement for Enhanced Control of Resist and Etch CDs", to appear in IEEE Transactions on CAD.

P. Gupta and N. Mangal, "Operation Scheduling in Reconfigurable Computing Environment", *Proc. VLSI Design and Test Workshop*, 1999.

P. Gupta, N. Mangal and C. P. Ravikumar, "Task Partitioning Between a General Purpose Microprocessor and Reconfigurable Hardware", *Proc. ACM International Symposium on FPGAs*, 2001.

P. Gupta, J. Abraham and R. A. Parekhji, "Improving Path Delay Coverage in Embedded Cores - Methodology and Experiments", *Proc. Texas Instruments Symposium on Test*, 2001.

Y. Cao, P. Gupta, A. B. Kahng, D. Sylvester and J. Yang, "Design Sensitivities to Variability: Extrapolation and Assessments in Nanometer VLSI", *Proc. IEEE ASIC/SoC Conference*, 2002.

P. Gupta, A. B. Kahng and S. Mantik, "Routing-Aware Scan Chain Ordering", *Proc. ACM/IEEE Asia and South Pacific Design Automation Conference*, 2003.

Y. Chen, P. Gupta, and A. B. Kahng, "Performance-Impact Limited Dummy Fill Insertion", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2003.

P. Gupta and A. B. Kahng, "Quantifying Error in Dynamic Power Estimation of CMOS Circuits", *Proc. IEEE Intl. Symposium on Quality Electronic Design*, 2003.

P. Gupta, A. B. Kahng and S. Mantik, "A Proposal for Routing-Based Timing-Driven Scan Chain Ordering", *Proc. IEEE Intl. Symposium on Quality Electronic Design*, 2003.

P. Gupta, A. B. Kahng, D. Sylvester and J. Yang, "A Cost-Driven Lithographic Correction Methodology Based on Off-the-Shelf Sizing Tools", *ACM/IEEE Design Automation Conference*, 2003.

Y. Chen, P. Gupta and A. B. Kahng, "Performance-Impact Limited Area Fill Synthesis", *ACM/IEEE Design Automation Conference*, 2003.

P. Gupta, A. B. Kahng, I. I. Mandoiu, and P. Sharma, "Layout-Aware Scan Chain Synthesis for Improved Path Delay Fault Coverage", *Proc. ACM/IEEE Intl. Conference on Computer-Aided Design*, 2003.

P. Gupta and A. B. Kahng, "Manufacturing-Aware Physical Design", *ACM/IEEE Intl. Conference on Computer-Aided Design*, 2003 (embedded tutorial).

D. Sylvester, P. Gupta, A. B. Kahng, and J. Yang, "Toward Performance-Driven Reduction of the Cost of RET-Based Lithography Control", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2003.

P. Gupta and A. B. Kahng, “Wire Swizzling to Reduce Delay Uncertainty Due to Capacitive Coupling”, *Proc. ACM/IEEE Intl. Conference on VLSI Design*, 2004.

P. Gupta, F.-L. Heng, M. Lavin, “Merits of Cellwise Model-Based OPC”, *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2004.

F.-L. Heng, P. Gupta, R. L. Gordon, K. Lai and J. Lee, “ Taming Focus Variation in VLSI Design”, *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2004.

P. Gupta, A. B. Kahng, Y. Kim and D. Sylvester, “Investigation of Performance Metrics for Interconnect Stack Architectures”, *Proc. ACM Intl. Workshop on System Level Interconnect Prediction*, 2004.

P. Gupta and F.-L. Heng, “Toward a Systematic-Variation Aware Timing Methodology”, *Proc. ACM/IEEE Design Automation Conference*, 2004.

P. Gupta, A. B. Kahng, P. Sharma and D. Sylvester, “Selective Gate-Length Biasing for Cost-Effective Runtime Leakage Reduction”, *Proc. ACM/IEEE Design Automation Conference*, 2004.

P. Gupta, A. B. Kahng, D. Sylvester and J. Yang, “Toward a Methodology for Manufacturability Driven Design Rule Exploration”, *Proc. ACM/IEEE Design Automation Conference*, 2004.

P. Gupta, A. B. Kahng, C.-H. Park, P. Sharma, D. Sylvester and J. Yang, “Joining the Design and Mask Flows for Better and Cheaper Masks”, *Proc. SPIE BACUS Symposium on Photomask Technology and Management*, 2004.

P. Gupta, A. B. Kahng, D. Sylvester and J. Yang, “Performance-Driven OPC for Mask Cost Reduction”, *Proc. IEEE International Symposium on Quality Electronic Design*, 2005.

P. Gupta, A. B. Kahng and C.-H. Park, “Detailed Placement for Improved Depth of Focus and CD Control”, *Proc. ACM/IEEE Asia and South Pacific Design Automation Conference*, 2005.

P. Gupta, A. B. Kahng and P. Sharma, “A Practical Transistor-Level Threshold Voltage Assignment Methodology”, *Proc. ACM/IEEE International Symposium on Quality Electronic Design*, 2005.

P. Gupta, A. B. Kahng and C.-H. Park, “Manufacturing-Aware Design Methodology for Assist Feature Correctness”, *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2005.

F.-L. Heng, P. Gupta and J.-F. Lee, “Through Process Layout Quality Metrics”, *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2005.

P. Gupta, A. B. Kahng, C.-H. Park, K. Samadi and X. Xu, “Topography-Aware Optical Proximity Correction for Better DOF margin and CD Control”, *Proc. SPIE Photomask and Next-Generation Lithography Mask Technology, Japan*, 2005.

P. Gupta, A. B. Kahng, Y. Kim, and D. Sylvester, “Self-Compensating Design for Focus Variation”, *Proc. ACM/IEEE Design Automation Conference*, 2005.

P. Gupta, A. B. Kahng and C.-H. Park, “Enhanced Resist and Etch CD Control by Design Perturbation”, *Proc. BACUS Symposium on Photomask Technology and Management*, 2005.

P. Gupta, A. B. Kahng, S. Muddu, O. S. Nakagawa, C.-H. Park, “Modeling OPC Complexity for Design for Manufacturability”, *Proc. BACUS Symposium on Photomask Technology and Management*, 2005.

Y. Zhang, R. Gray, O. S. Nakagawa, P. Gupta, H. Kamberian, G. Xiao, R. Cottle, and C. Progler “Interaction and Balance of Mask Write Time and Design RET Strategies”, *Proc. SPIE Photomask and Next-Generation Lithography Mask Technology, Japan*, 2005.

P. Gupta, A. B. Kahng and C.-H. Park, “Improving OPC Quality Via Interactions Within the Design-to-Manufacturing Flow”, *Proc. SPIE Photomask Japan*, 2005.

P. Gupta, A. B. Kahng, O. S. Nakagawa and K. Samadi, “Closing the Loop in interconnect Analyses and Optimization: CMP Fill, Lithography and Timing”, *Proc. 22nd Intl. VLSI/ULSI Multilevel Interconnection (VMIC) Conference*, 2005.

P. Gupta and A. B. Kahng, “Efficient Design and Analysis of Robust Power Distribution Meshes”, *Proc. ACM/IEEE Intl. Conference on VLSI Design*, 2006.

P. Gupta, A. B. Kahng, S. Nakagawa, S. Shah, and P. Sharma “Lithography Simulation-Based Full-Chip Design Analyses”, *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2006.

P. Gupta, A. B. Kahng, Y. Kim, S. Shah, and D. Sylvester “Modeling of Non-Uniform Device Geometries for Post-Lithography Circuit Analysis”, *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2006.

P. Gupta, A. B. Kahng, S.V. Muddu, and S. Nakagawa “Modeling Edge Placement Error Distribution in Standard-Cell Library”, *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2006.

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”, *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2006.

P. Gupta, A. B. Kahng and S. Shah, “Standard-Cell Library Optimization for Leakage Reduction”, *Proc. ACM/IEEE Design Automation Conference*, 2006.

ABSTRACT OF THE DISSERTATION

On Compensation of Systematic Manufacturing Variations in Physical Design

by

Puneet Gupta

Doctor of Philosophy in Electrical Engineering (Computer Engineering)

University of California, San Diego, 2007

Professor Andrew B. Kahng, Chair

Scaling of physical dimensions faster than the optical wavelengths or equipment tolerances used in the manufacturing line has led to increased process variability. This in turn has led to unpredictable design, unpredictable manufacturing, and low yields. The result of these physical variations is variation in circuit metrics such as performance and power.

Variations can be either systematic (e.g., metal dishing, lithographic proximity effects, etc.) or random (e.g., material variations, dopant fluctuations, etc.). The former can be modeled and predicted while random variations are inherently unpredictable. There are several pattern-dependent process effects which are systematic in nature. These can be compensated during physical design to aid manufacturability and hence improve yield. This thesis focuses on ways to mitigate the impact of systematic variations on design and manufacturing by establishing a bidirectional link between the two. The motivations for doing so are improved yield and manufacturability as well as reduced design guardband and cost.

To improve manufacturability, we propose a detailed placement perturbation technique for improved depth of focus and process window. The technique facilitates downstream insertion of scattering bars and etch dummy features in the resolution enhancement process, reducing inter-cell forbidden pitches almost completely. We propose a systematic variation aware timing analysis methodology to reduce timing pessimism. The Proposed self-compensated design techniques achieve circuit robustness to focus variation. We also propose the use of small gate-length biases to reduce leakage power and leakage power variability. To reduce design guardbanding, we have proposed a methodology for power/performance analyses of the design based on lithography-simulation output. We also give the first method for performance-impact limited metal fill inser-

tion. Finally, to reduce mask cost, mask data volume, as well as mask data preparation time, we propose a novel design-aware optical proximity correction (OPC) methodology.

I

Introduction

Yield is defined as the ratio of the number of products that can be sold to the number of products that can be manufactured. The estimated typical cost of modern 300 mm or 12 inch wafer 0.13 μm process fabrication plant is $2-4 billion. The typical number of processing steps for a modern integrated circuit is more than 150. Typical production cycle-time is over 6 weeks. Individual wafers cost multiple thousands of dollars. Given such huge investments, consistent high yield is necessary for faster time to profit.

I.A Taxonomy of Yield Loss

Total yield for an integrated circuit Y_{total} can be expressed as follows [113].

$$Y_{total} = Y_{line} \times Y_{batch} \tag{I.1}$$

Here Y_{line} denotes line yield or wafer yield which is the fraction of wafers that survive through the manufacturing line. Y_{batch} is the fraction of integrated circuits which on each wafer which are fully functional at the end of the line. Steep yield ramp means quicker path to high batch yield and hence volume production. Earlier volume production means higher profitability for the semiconductor manufacturer in today's market with time-to-market pressures.

Y_{batch} can be further classified based on either type of defect or of failure. Failure-type taxonomy is as follows.

- *Catastrophic Yield Loss.* These are functional failures such as open or short circuits which cause the part to not work at all. Extra or missing material particle defects are the primary causes for such failures.

- *Parametric Yield Loss.* Here the chip is functionally correct but it fails to meet some power or performance criteria. Parametric failures are caused by variation in one or set of circuit parameters, such that their specific distribution in a design makes it fall out of specifications. For example, parts may function at certain VDD, but not over the whole required range. Parametric failures may be caused by process variations. Several kinds of integrated circuits are *speed-binned* (i.e.,grouped by performance). A common example of such class of designs is microprocessors wherein lower performance parts are priced lower. The other class is typical ASICs which cannot be sold if the performance is below a certain threshold (for example, due to compliance with standards). In the latter case, there can be significant performance-limited yield loss which is why such circuits are designed with a large guardband. In the former case too, there can be significant dollar value loss even if there is little yield loss.

Additionally, there is also testing-related yield loss as no testing process can detect all possible faults (and potential faults).

Defect types can be classified as follows.[1]

- *Random Defects.* These are randomly distributed faults such as particle contamination.

- *Systematic Defects.* These kind of defects are predictable. Example sources include CMP (Chemical Mechanical Polishing) and photoresist pattern collapse.

It is important to understand that both random and systematic defects can cause parametric or catastrophic yield loss. For example, lithographic variation which is typically systematic and pattern-dependent can cause catastrophic line-end shortening leading to the gate (polysilicon over diffusion) not forming and hence a functional failure.

[1] A similar taxonomy is typically used for process variations as well. The terms defects and variations are used interchangeably in literature. One common distinction between the two terms is the exclusion of the particle defects from variations.

A less drastic rendition of lithographic variation is gate-length variation causing gates on critical paths to speed up too much, which leads to hold-time violations under certain voltage and temperature conditions. Systematic mechanism limited yield loss is projected to be the dominant source of yield loss in current and future technology generations [113].

I.B Sources of Yield Loss

As mentioned earlier, yield loss can result from either systematic or random defects. Contamination related spot defects are not the focus of my work. In this section we focus our attention to variations. There are several ways to classify variations depending on the axis:

- Process vs. Environmental. Variation occurring during circuit operation (e.g., temperature, power supply, etc.) are environmental in nature while those occurring during the manufacturing process (e.g., mask misalignment, stepper focus, etc.) are physical. We will focus only on process variations.

- Systematic vs. Random. Systematic variations (e.g., metal dishing, lithographic proximity effects, etc.) can be modeled and predicted while random variations (e.g., material variations, dopant fluctuations, etc.) are inherently unpredictable.

- Inter-die vs. Intra-die. Depending on the spatial scale of the variation, it can be classified as die-to-die (e.g., material variations) or within-die (e.g., layout pattern-dependent lithographic variation). Inter-die variations correspond to variation of a parameter value across nominally identical die. Such variations may be die-to-die, wafer-to-wafer or even lot-to-lot. Inter-die variations are typically accounted for in design, by shift in the mean of a parameter value. Intra-die variations on the other hand correspond to parameter fluctuations across nominally identical circuit elements such as transistors. Intra-die perturbations are usually accounted for in design by guardbanding and prevention. Variation compensation in design is further discussed in the next section.

An interesting point to note here is the level of abstraction for sources of variation. For logic designers, variation may be caused by cell delay or transistor delay

changes. Such modeling is evident, for example, in most statistical timing analysis tools (e.g., [49, 166, 88]). For circuit designers, the level of abstraction may go down to (say) transistor gate-length variation which leads to cell or transistor delay variation. Going further down, a lithographer may attribute critical dimension (CD) variation to focus variation which may be further blamed on wafer flatness imperfections.

Variation in process conditions can manifest itself as dimensional variations or material variations. Dimensional variations include the following.

- Lateral dimension variation. Across chip linewidth variation or ACLV is one of the biggest contributors to parametric variation. In this category important causes of parametric and functional failure are gate-length variation, line-end pullback and contact or via overlap. Lithography and etch processes are the biggest culprits for ACLV variations. Such variations are largely systematic and layout pattern-dependent.[2] With scaling geometries, even small variations in dimensions can be detrimental to circuit performance. For example, line edge roughness (LER) is projected to be a big concern for 32nm device performance [114, 92].

- Topography variation. Dielectric erosion and metal dishing caused by chemical mechanical polishing (CMP) processes is one of the biggest contributors to interconnect failures. In front-end of the line (FEOL), imperfect STI (Shallow Trench Isolation) CMP process is a sample cause of topographic variation. Topographic variation not only results in interconnect resistance and capacitance variation but, by virtue of acting as defocus for lithographic manufacturing of subsequent layers, results in linewidth variation [115].

Several processing steps during the manufacture of deep sub-micron integrated circuits can result in material parameter perturbations. Besides material purity variations, such variations can also be caused by perturbations in the implantation or deposition processes. An important example of material variation is discrete dopant fluctuation. Random placement of atoms at discrete location in the channel can cause V_{th} variation. With number of dopant atoms going down to few hundred in sub-100 nm devices, random dopant fluctuation is becoming an important source of variation.

[2] Lateral dimension variation is typically mitigated on the manufacturing side by resolution enhancement techniques (RETs) such as optical proximity correction (OPC).

The result of these physical variations is variation in circuit metrics like performance and power. International Technology Roadmap for Semiconductors (ITRS) projects as much as 15% slow-down in design signoff delay by the year 2014. Leakage and leakage variability is an even bigger problem due to exponential dependence of leakage power on physical dimensions like gate-oxide thickness and gate-length, as well material properties like dopant concentration. 30X variation in leakage in microprocessor has been noted by [116]. According to ITRS projections, containing V_{th} variability to within 58%, circuit performance variability to within 57% and circuit power variability to within 59% is a "red-brick" (i.e., no known solutions). On the BEOL (Back End of the Line) side, varying electrical parameters include via resistance as well as wire resistance and capacitance.

I.C Common Methods for Yield Optimization

Aggressive technology scaling has made process variation control from purely manufacturing perspective very tough. Design-related yield losses have been projected to increase which implies that greater cooperation between physical design and process communities is necessary. Yield optimization methods work with the "measure, model and mitigate" flow. Measurements are usually done by targeted test structures which are measured on silicon for physical parameters like linewidth and thickness as well as electrical parameters like sheet resistance and transistor saturation current. Models of process extracted from such test-structure measurements are usually abstracted to simpler models or a set of rules for physical design and verification tools to use. In this section, we will briefly discuss the evolution of yield optimization physical design techniques.

I.C.1 Design Rules

Abstraction of manufacturing constraints into a set geometric of constraints or design rules for the layout designers to follow have traditionally been foundry's main method to ensure high probability of correct fabrication of integrated circuits. Typical design rules are constraints on width, spacing or pattern density. Origins of design rules

lie in various manufacturing steps such as lithography, etch, implant, CMP, etc. Other factors influencing design rule values include preserving scaling, area overhead, layout migratability and ability of design tools and flows to handle them.

Manufacturability implications of technology scaling have led to three major trends in design rules:

- More complicated rule sets. The sheer number of design rules has been growing at a rapid pace with every technology generation. More process constraints have required new kinds of rules [167, 168]. This has made physical verification, routing as well as custom layout very difficult and time consuming tasks.

- Restrictive design rules. To cope with sub-100 nm manufacturability concerns where manufacturing equipment is not keeping pace with feature scaling, radically restraining layout options has been proposed as a viable option [31]. One common restriction is to enforce regularity in layout which aids printability. An example of such a rule is allowing only one or two pitches on the polysilicon layer.

- DFM rules. Most 90 nm and 65 nm design rule manuals include a separate set of non-minimum design rules. These design rules if obeyed by the layout, enhance its manufacturability. For example, the minimum metal-via enclosure can be 20nm while the corresponding DFM rule can be 30nm. The increased enclosure can reduce chances of loss of contact between metal route and via at the cost of increased routing area.

Though design rules have served the industry well in the past as the abstraction layer, inadequacy and sub-optimality of such yes/no rules has led to a slow but steady adoption of model-based checking methods.

I.C.2 Corner-Based Design Analysis

Traditionally, static timing and power analysis tools have relied on two or more *corners* of process, voltage and temperature (PVT). We will not discuss operating variations such as voltage fluctuations and temperature gradients here. Timing corners are typically specified as slow (S), typical (T) or fast (F). Thus, SS represents a process corner

with slow PFET and slow NFET behavior. The common performance analysis process corners are (TT, SS, FF, SF, FS). Similarly, interconnect parasitics are extracted at multiple (usually two) corners. Usually hold time violations are checked at the FF corner and setup time violations are checked at the SS corner. Similarly, interconnect parasitics can also have typical, minimum and maximum values. The rationale for corner-based analyses lies in the fact that ensuring correct operation of the design at the PVT extrema ensures correct operation throughout the process and operation range. This assumption though not strictly correct, usually holds well in practice. Corner-based analysis enables pessimistic but deterministic analysis and optimization of designs. Most modern physical design algorithms rely on corner based design being acceptable. Sub-100 nm process issues (especially variability) have led to the following trends in corner-based design analysis and optimization.

- More corners. As more complicated process effects emerge and as a result of non-monotone dependence of delay on many of the process parameters, the number of PVT corners at which a design needs to be signed off is increasing.

- On-chip Variation (OCV) analysis. To model within-die variation in static timing tools implicitly analyze clock paths and data paths at separate corners. For example, for setup time analysis, the launching clock path may be analyzed at a slow corner while the capturing clock is analyzed at a fast corner and the data path is analyzed at the slow corner. This in essence tries to model the worst-case impact of On-chip variation. Additional techniques such as common path pessimism removal (CPPR) which figures out the shared logic between launching and capturing paths to avoid pushing them to different corners, are used to reduce the inherent pessimism in OCV analysis.

Though the runtime overhead of ever-increasing number of corners, the excess pessimism in corner-based analysis and fear of missing some corners in a high process-variability regime has led to an increasing interest in statistical analysis tools, corner-based design deterministic design optimization still remains the mainstay of commercial parametric yield optimization.

In addition to the above design-side methods, the foundry constantly works to tune the process and reduce its variability. The three measures of quality for the process which are commonly used are $D0$ (a measure of average defect density), Cp (a measure of process variation relative to tolerable variation in design specifications) and Cpk (a measure of process centering). Process engineers continuously work to decrease $D0$ and increase Cp and Cpk. From a process mindset Cp is a measure of random variability inherent in the process while Cpk is a measure of systematic variability which can be assigned to causes and potentially fixed. In this thesis we will focus on design methods which increase Cp and Cpk.[3]

I.D Systematic Variations

As mentioned earlier, variations can be either systematic or random. Random variations can be optimized and analyzed statistically (or in a worst-case sense). Any systematic variation that cannot be modeled sufficiently is typically treated as a random variation. If the modeling is worst-case, this can be overly pessimistic. If the modeling is statistical, it can be overly optimistic in certain scenarios and overly pessimistic in others. For this reason, it is important to model and compensate systematic variations *before* any statistical optimization is done for the random components of the variability.

There are several pattern-dependent process effects which are systematic in nature. These can be compensated for during physical design to aid manufacturability and hence improve yield. The biggest contributors in this bucket are CMP and photolithography.

I.D.1 Chemical Mechanical Polishing (CMP) Effects

CMP is the chemical-mechanical polishing of wafer surface using a rotary pad and chemical slurry (see Figure I.1). In modern processes, copper, being softer than the dielectric oxide, gets polished quicker. As a result, the resulting wafer topography is dependent on underlying metal density. Varying topography means varying dielectric height and metal thickness resulting in varying interconnect resistance and capacitance.

[3] Note that the physical parameter variation itself cannot be controlled by design. Here we are talking in terms of variability in design metrics like power and performance.

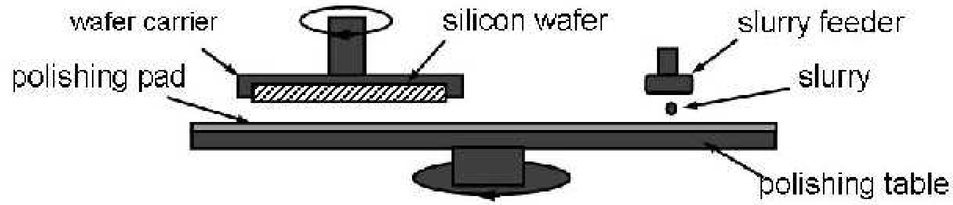


Figure I.1: The CMP process.

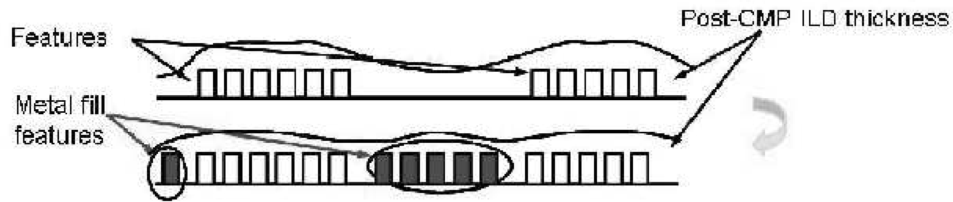


Figure I.2: Impact of metal fill on topography.

This performance variation makes prediction and compensation of CMP-related variation very important for design predictability. This density-dependent topography variation is predictable but the models are usually too complex to be used in physical design. To reduce the CMP-induced topography variation, modern design flows insert dummy metal fill (see Figure I.2) features to make metal density as uniform as possible. Dummy fill synthesis is a mandatory step for all metal layers in sub-100 nm design flows. This step can be done post-tapeout inside the foundry, as part of physical verification or as part of the router. In any of the cases, conventional metal fill is not aware of its timing impact. The added capacitance and hence its impact on design signoff remains unpredictable.[4]

I.D.2 Photolithography Effects

While CMP is the biggest source of systematic variation in the back-end-of-the-line (BEOL) process, photolithography fills that slot for front-end-of-the-line (FEOL) process. Lithography is the process of transfer of design patterns from mask or reticle onto the wafer.

The mask is usually written directly using a variable shaped electron-beam (VSB) mask writer. The VSB writer "shots" on layout features[5] which makes the mask

[4]Parasitic extraction of floating metal fill is slow and inaccurate and therefore, rarely part of design flows.

[5]Here layout features referred to are simple trapezoids into which the entire layout is fractured.

write-time very sensitive to the layout complexity. Time on these expensive mask writers is expensive which makes mask cost a sizeable components of low-volume IC manufacturing costs.

UV illumination is used to process a photoresist on wafer surface to form the required features. The wavelength of the light used is much smaller than the required resolution ($\lambda = 193nm$ is used for 90 nm and 65 nm technology nodes). To enable this subwavelength lithography, several resolution enhancement techniques (RETs) are used by lithography engineers. Four commonly used RETs are as follows.

- Optical proximity correction sizes layout (OPC). geometries to reduce the printing error. OPC can be either rule-based or model-based (i.e. with an embedded lithography simulator) and is applied to almost all fabrication layers in sub-100 nm processes. OPC is a compute-intensive process and increases the layout complexity by over 10X.

- Off-Axis Illumination (OAI). OAI involves use of tilted illumination to improve process window. Unfortunately, any given off-axis configuration maximizes depth of focus (DoF)[6] for one pitch while worsening it for others (DoF for pitch twice the value of the optimal OAI pitch is worsened the most). The pitch values for which the process window is too small are typically referred to as "forbidden pitches" by the lithographers.

- Phase Shift Mask (PSM). PSM involves adding phase shifters to the mask around the layout feature to leverage destructive interference for reducing the minimum resolvable dimension [15]. Though attenuating PSM is commonly used, the stronger version of PSM, namely, alternating PSM is much less common due to increase in number of masks as well as the layout constraints it imposes [14].

- Sub-Resolution Assist Features (SRAFs). SRAFs or scattering bars are narrow lines inserted near isolated geometries so that they behave lithographically like nested features, thus enhancing process overlap window between isolated and dense lines.

[6]Lithographic process window is usually measured by *exposure latitude* i.e., the tolerable deviation in illumination intensities and *depth of focus* i.e., the tolerable deviation of lens focus.

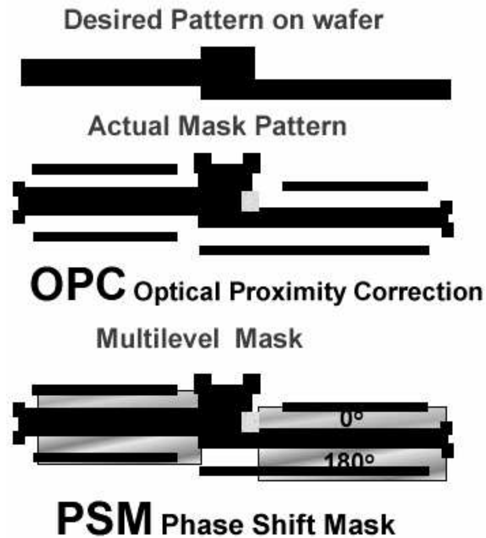


Figure I.3: A picture showing OPC and alternating PSM. Also notice the SRAFs next to the OPC'd feature.

Figure I.3 shows some of the above RETs.

The effectiveness of above RETs can be helped immensely by smarter physical design and layout. As discussed earlier, the methods for making layout amenable to RETs and hence more manufacturable has centered around more complex design rules or radically restricted design rules. In addition, there are a few research publications which have focused on more explicit accounting of systematic variations in physical design. [6] proposes a detailed placement perturbation technique to legalize the placement with respect to forbidden pitches. Similarly, a route-fix technique is described in [8] to remove BEOL forbidden pitches. A more explicit lithography-simulation-driven approach to routing is followed in [9]. Similarly, for planarization, design rules specifying window-based minimum and maximum density bounds have been used [126]. The density rules are becoming increasingly complex with multiple-window and multiple-layer constraints.

Another aspect is bringing systematic-variation awareness into timing and power analyses as well as optimization. Traditionally, corner based methods have been the only way to accomplish this. In the past few years, abstracted models have started showing up in commercial EDA tools. An example here would be awareness of metal density in parasitic extraction tools. Simple density-based look-up tables are used to model

post-CMP topography variation. Extraction of metal fill geometries is also a tricky task because of the sheer number of them as well as due to the fact they are usually floating. A simplification by replacing metal fill by an equivalent high-k dielectric has been proposed in [136]. On the lithography front, recently there has been a lot of interest in modeling impact of RET and lithographic variation induced errors on timing and power. Works such as [44, 104] use simple look-up table based abstracted 1D models for layout-dependent CD variation to perform a variation-aware timing analysis. A more explicit approach of feeding lithography simulation results back to a timing/power analysis flow is followed in more recent works such as [111, 112].

Prediction and compensation of systematic variations has traditionally been done by the manufacturing process with only simple guardbanded abstractions (e.g. design rules) being passed on to the designers. Increasing magnitude and impact on design metrics of these variations coupled with the inability of manufacturing equipment and process tricks to deal with them, makes systematic variations a big yield limiter in sub-100 nm technologies. More active participation of design in systematic variation compensation is essential to maintain acceptable levels of manufacturing yield and design predictability as technology scales.

I.E This Thesis

Traditionally, design rules have been the method to optimize for systematic variation. Recently more explicit mitigation of impact of systematic variation on circuit power and performance has been studied. My research focuses on ways to mitigate impact of systematic variations on design and manufacturing by establishing a bidirectional link between the two. This thesis addresses the following issues in particular.

- *Design Guardbanding.* If systematic variations are not explicitly modeled in design, they are lumped into the random variation bucket. As a result, there is overdesign for them assuming a worst-case impact. Such guardbands occur in both timing and power and can make the final chip-signoff tougher than it need be. More importantly, the guardband may be inserted at the wrong places for lack of understanding of systematic nature of the impact of the variations.

- *Variability.* Compensating for variations which are systematic or have a systematic impact on circuit characteristics such as performance and power reduces total variability in these electrical characteristics even if the physical dimension variability remains the same. Reduced electrical variability results in better parametric yield as well as better quality products.

- *Cost.* Mitigating all systematic variability by various process correction (e.g., Optical Proximity Correction (OPC)) methods can be an expensive task both in terms of turnaround time as well as monetary cost. Realizing where this compensation is important and where it is not from a design perspective can bring down this cost of correction by a significant amount.

- *Manufacturability.* If sources of systematic pattern-dependent variation are known, it is possible to perturb the design layout so as to reduce the occurrence of "bad patterns". This leads to improved functional as well as parametric yield besides making the process correction tasks "easier".

Chapter II proposes perturbation of detailed placement of standard cells to redistribute whitespace in order to remove the so-called "forbidden pitches" in the layout and make the layout more amenable to SRAF and etch dummy insertion downstream. This increases manufacturability of the design, almost completely eliminating edge placement errors at worst defocus.

Chapter III proposes timing analysis methodology which is aware of interaction between layout and lithographic defocus, reducing uncertainty by up to 40%. Moreover, it proposes a novel self-compensated design methodology for compensating random focus variations.

Chapter IV proposes use of very small increases in gate-length to reduce leakage and leakage variability. Such biases can be layout-preserving and we proposed sensitivity-based algorithm for selective gate-length biasing in standard-cell designs under timing constraints achieving up to 30% reduction in leakage and 40% reduction in leakage variability.

Chapter V introduces a full-chip timing and power analysis methodology including both wires and gates to analyze litho-simulated contours. We have also proposed the

first model for non-rectangular channels which accounts for narrow width effects.

Chapter VI develops a novel minimum cost of correction methodology to determine the level of correction of each layout feature such that prescribed parametric yield is attained with minimum RET cost. Designs adopted with this methodology achieves 17%-24% MEBES data volume reduction and 6%-41% OPC runtime reduction.

Chapter VII proposes heuristics for metal fill insertion that show substantial improvements in setup timing slack as well as total added delay (up to 90% improvement) while maintaining identical quality of the layout density control.

I.F Acknowledgments

Chapter I is in part a reprint of "Yield Analysis and Optimization", *The Handbook of Algorithms for VLSI Physical Design Automation*, CRC Press, to be published. I would like to thank my coauthor Dr. Evanthia Papadopoulou.

II

Dealing with FEOL Forbidden Pitches in Detailed Placement

Across-chip linewidth variation (ACLV) induced by photolithography and etch processes has been a major barrier in ultra-deep submicron manufacturing. Photolithography has been a key enabler of the aggressive IC technology scaling implicit in Moore's Law. Minimum feature sizes have outpaced the introduction of advanced lithography hardware solutions so that gate-length and CD tolerances prescribed in the 2003 International Technology Roadmap for Semiconductors (ITRS) [2] are extremely difficult to achieve. As a result, resolution enhancement techniques (RETs) such as optical proximity correction (OPC) [22], phase-shift masks (PSM) [15], and OAI are being pushed ever closer to fundamental resolution limits [30]. Combinations of these techniques can provide certain advantages for lithography manufacturing, e.g., OAI and OPC, together with SRAF, achieve enhanced CD control and focus margin at minimum pitch.

However, when OAI is used, there will always be pitches for which the angle of illumination works with the angle of diffraction to produce a bad distribution of diffraction orders in the lens. These pitches are called *forbidden pitches* because of their lower printability, and designers should avoid such pitches in the layout. Forbidden pitches consist of Horizontal (H-) and Vertical (V-) forbidden pitches, depending on whether they are caused by interactions of poly geometries in the same cell row or in different cell rows, respectively. The resulting *forbidden pitch problem* for the manufacturing-critical

poly layer must be solved before detailed routing. Since detailed routing works on fixed placement except some small placement ECOs as required, detailed routing "locks in" the poly layer layout. At the same time, we wish to address the forbidden pitch problem as late as possible, to avoid extra rework upon modification of the manufacturing process recipe. In this chapter, we first describe a novel dynamic programming-based algorithm for *AFCorr* (assist feature Correctness), which uses flexibility in detailed placement to avoid all possible H- and V-forbidden pitches and the manufacturing uncertainty that they cause.

Etch dummy features are introduced into the layout to reduce the CD distortion induced by etch proximity. The etch dummies are placed at the outside of active layers so that leftmost and rightmost gates on active-layer regions are protected from ion scattering during the etch process. However, etch dummy rules conflict with SRAF insertion because each of the two techniques requires specific spacings from poly.

The etch dummies are placed at the outside of active layers so that leftmost- and rightmost-gates on active-layer regions are protected from ion scattering during the etching process. However, etch dummy rules conflict with SRAF insertion because each of the two techniques requires specific spaces from poly. In such a regime, the assist feature correct placement methodology must consider assist feature and etch dummy correction. In this chapter, we also present a novel SRAF-aware etch dummy insertion method (*SAEDM*) which applies flexible etch dummy rules according to the distance from active edge to leftmost (or rightmost) poly. As a result, the layout will be more conducive to assist feature insertion after etch dummy features have been inserted. Finally, we introduce a dynamic programming-based technique for etch-dummy correctness (*EtchCorr*) which can be combined with the SAEDM in detailed placement of standard-cell designs.

II.A Assist Feature Correctness

II.A.1 RET and Layout Impact

The extension of optical lithography beyond the quarter-micron regime has been enabled by a number of resolution enhancement techniques. These RETs address

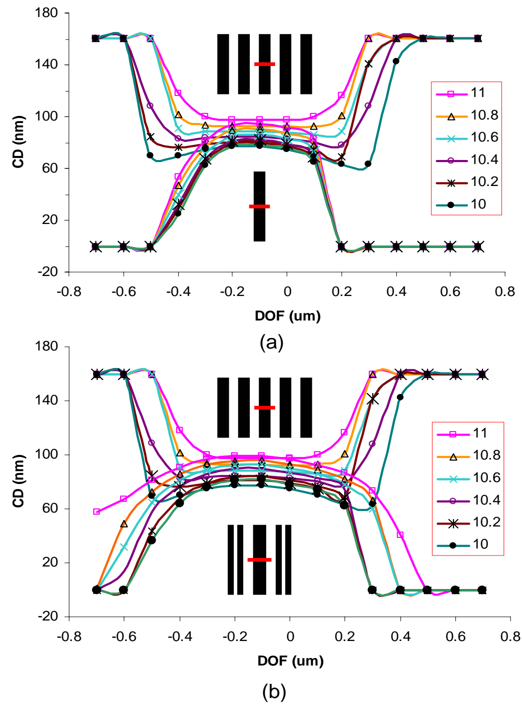


Figure II.1: Comparison of Bossung plots between dense and isolated lines: (a) results of Bias OPC and (b) results of SRAF OPC.

the three available degrees of freedom in lithography, namely, aperture, phase, and/or pattern uniformity [18, 19]. However, the adoption of different RETs dictates certain tradeoffs with various aspects of process and performance [3].

Off-axis illumination (OAI) brings light to the mask at an oblique angle. As the angle of diffraction through certain aperture shapes matches a given pitch, higher-order pattern information can be projected on the pupil plane as determined by the numerical aperture (NA) of the illumination system. This technique enables the certain pitches on the mask to obtain higher resolution and extended focus margin. However, other pitches beyond the optimum angle will have a *lower* process margin compared to conventional illumination (i.e., with a circular aperture). Since OAI is an essential technique in current lithography, these other pitches should be forbidden, and their avoidance is a new challenge for physical design automation. OPC is the deliberate and proactive distortion of photomask shapes to compensate for systematic and stable patterning inaccuracies. *Bias OPC*, the most common and straight-forward application of OPC, has proved to be a useful technique for matching photoresist edges to layout edges with essentially

a layout sizing technique. However, bias OPC has limitations in enhancing process margins with respect to depth of focus and exposure dose. The Bossung plot[1] in Figure II.1 shows that bias OPC is not sufficient to reduce the CD difference between isolated and dense patterns with varying focus and exposure dose. The CD distortion in the isolated pattern is usually a problem since lithography and RET recipes are not tuned or optimized for isolated lines [16]. The *SRAF OPC* technique combines pattern biasing with assist feature insertion to compensate for the deficiencies of bias OPC. SRAFs (or, Scattering Bars (SB)), which are extremely narrow lines that do not actually print on the wafer, modify the wavefront and allow the lens pupil to receive higher-order pattern information. The SRAFs are placed adjacent to primary patterns, such that a relatively isolated primary line behaves more like a dense line. This works well for bringing the lithographic performance of isolated and dense lines into agreement. The DOF margin of the isolated line as shown in Figure II.1(b) is considerably improved from that shown in Figure II.1(a), and a larger overlap of process window[2] between dense and isolated lines is achieved.

The key observation is that the SRAF technique places more constraints on the spacing between patterns. SRAFs can be added whenever a poly line is sufficiently isolated, but certain minimum assist-to-poly and assist-to-assist spacings are required to prevent SRAFs from printing in the space [14]. We now briefly review previous works related to forbidden pitches and their design implications. Socha et al. [21] observe that under more aggressive illumination schemes such as annular and quasar illumination, some optical phenomena become more prominent, most notably the forbidden pitch phenomenon. Shi et al. [20] give a theoretical analysis of pattern distortion in forbidden pitches, due to destructive light field interference. Although SRAFs are an effective method to collect high-order diffraction on the entrance pupil plane of a projection lens [17], Shi et al. report that incorrect SRAF placements around a given main feature can actually degrade the process latitude of that feature. A number of previous works have

[1]The Bossung plot shows multiple CD versus defocus curves at different exposure doses, and has been a useful tool to evaluate lithographic manufacturability. The common process window between dense and isolated patterns is an increasingly important requirement to maintain CD tolerances in the subwavelength lithography regime.

[2]Process window is defined as the range of exposure dose and defocus within which acceptable CD tolerance is maintained.

proposed techniques to control forbidden pitches using optimization of optical conditions such as numerical aperture (NA) and illuminator aperture shape of OAI [24, 12, 13, 23]. However, if the assist feature insertion is not considered during layout, sizing of assist feature and adjustment of exposure dose must be applied. This will cause problems in mask inspection, as well as CD degradation. For instance, smaller SRAFs make mask inspection difficult and require higher-resolution inspection tools.

II.A.2 SRAF Rule and Forbidden Pitch Extraction

Lack of space may prohibit insertion of a sufficient number of SRAFs, and as a result patterns may violate CD tolerance through defocus. *Forbidden pitches* are pitch values for which the tolerance of a given target CD is violated. *Allowable pitches* are all pitches other than forbidden pitches. In this section, we summarize the criteria for SRAF insertion and forbidden pitch extraction considering a worst-defocus model. Our SRAF insertion rule is initially generated based on the theoretical background given in [20]. Positioning of SRAFs is then adjusted based on OPC results. Large CD degradation through-pitch increases pattern bias as model-based OPC is applied, and this requires trimming of the SRAF rule to guarantee better process margin and prevent the SRAFs from printing.[3] After applying SRAF OPC to test patterns with the best-focus model, OPC'ed pitch patterns are simulated with the worst-defocus model which will be described in detail later. This evaluation yields the forbidden pitches, considering maximum printability and manufacturability. The forbidden pitch rule is determined based on CD tolerance and worst defocus level, which are in turn dependent on requirements of device performance and yield. SRAF OPC restores printing when there is enough room for one scattering bar. But then larger pitches are forbidden until there is enough room for two scattering bars. We thus can extract a set of forbidden pitches which will be demonstrated later. In all of the work we report here, CD tolerance is assumed to be $\pm 10\%$ of minimum line width while the worst defocus level is assumed to be 0.5μm and 0.4μm for the 130 nm and 90 nm technology nodes, respectively. All of these results are summarized in Table II.1.

[3]More complicated approaches to SRAF rule generation may involve co-optimization of model-based OPC and SRAF insertion. We do not address such involved optimizations of OPC, since the focus of our work is OPC-aware design and not OPC itself.

II.A.3 AFCorr Placement Algorithm

In this subsection, we describe the proposed AFCorr placement perturbation algorithm for assist feature correction. Single orientation polysilicon geometries are becoming common for the current and future process generations. We consider the H-forbidden pitches within a cell row and the V-forbidden pitches between adjacent cell rows [6, 7]. In the present work, we treat the placement of a given cell row independently of all other rows, even though the cost function is calculated with respect to both H- and V-perturbations in order to avoid all forbidden pitches. Assuming that the spacings within the cell are assist-correct, then the only source of incorrect spacings between poly shapes for assist feature insertion is cell placement. Adjacent cells within the same standard-cell row as well as cells within adjacent cell rows which have shapes overlapping interact for this purpose. The vertical poly shapes (typically gates) at the left and right periphery of a cell which overlap with similar shapes in the neighboring cells in the row constitute the horizontal interaction. Similarly, horizontal poly shapes (typically field) at the top and bottom periphery of the cell which overlap with similar shapes in vertically adjacent cells (in adjacent rows) constitute the vertical interaction. In the following we describe the *single-row* AFCorr perturbation algorithm, which solves the 2D AFCorr problem one cell row at a time.

Let $C_{a,j}$ be a cell at the ath position in the jth row. To explain the interactions of border poly geometries, we adopt the following notations.

- **Horizontal polygon interaction:** Given a cell $C_{a,j}$, let $LP_{a,j}$ and $RP_{a,j}$ be the sets of valid poly geometries in the cell which are located closest to the left and right outlines of the cell, respectively. Only geometries with length larger than the minimum allowable length of SRAF features are considered. Define $s_{a,j}^{LP^i}$ to be the space between the left outline of the cell and the ith left border poly geometry. O_{gg}, O_{ff} and O_{gf} correspond to the length of overlapped area in the cases of gate-to-gate, field-to-field and gate-to-field-poly as shown in Figure II.2. In addition, c_{gg}, c_{ff}, and c_{gf} are proportionality factors which specify the relative importance of printability for gate and field-poly[4]. Typically, gate-poly geometries need to be

[4] Gate is the overlap region of polysilicon and diffusion. Field poly represents the rest area of polysilicon except the gate.

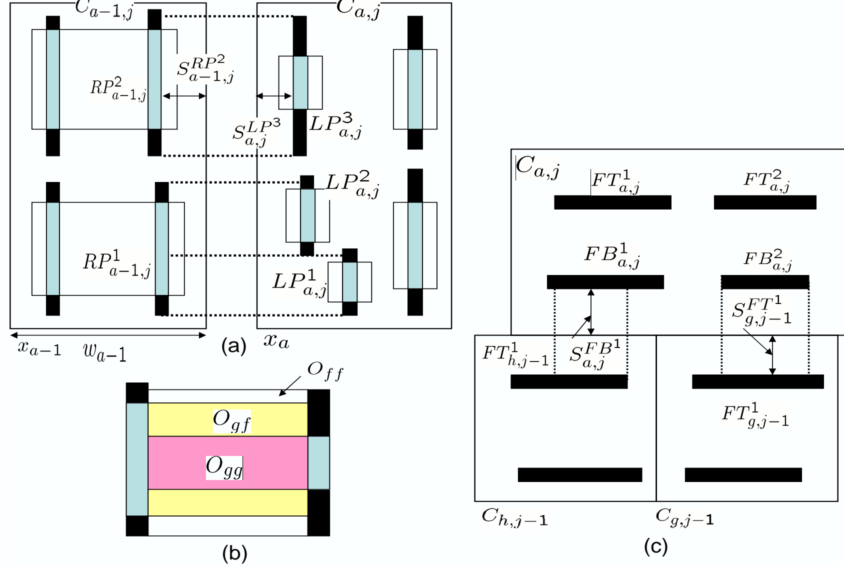


Figure II.2: (a) H-interactions of gate-to-gate, gate-to-field, and field-to-gate, (b) overlapped area of (a), (c) V-interactions of field-to-field-polys.

better controlled through process as they have more direct impact on performance. Therefore, a typical order of importance is $c_{gg} \geq c_{fg} \geq c_{ff}$.

- **Vertical polygon interaction:** Given a cell $C_{a,j}$, let $FB_{a,j}$ and $FT_{a,j}$ be the sets of valid field-poly geometries in the cell which are located closest to the bottom and top outlines of the cell, respectively. Define $s_{a,j}^{FB^i}$ ($s_{a,j}^{FT^i}$) to be the space between the bottom (and top) outline of the cell and the ith bottom (and top) border poly geometry. O_{ff} corresponds to the length of field-to-field overlap between horizontal geometries in adjacent cell rows.[5]

Assume an ordered set $AF = AF_1, \ldots, AF_m$ of spacings which are "assist-correct", i.e., if the spacing between two gate-poly shapes belongs to the set AF, then the required number of assist features can be inserted between the two poly geometries. For example in Figure II.7, the peaks of the CD correspond to AF_i. The acceptable CD tolerance range (e.g., 10%) results in a range of acceptable pitches starting at AF_i. AF is assumed to be sorted in increasing order. Note that the set AF may contain a number of spacings which correspond to varying SRAF widths. Let w_a denote the width of cell

[5]Gates are typically laid out in a single orientation. We assume this orientation to be vertical in this work.

HCost(a,b,a-1,i) of Cell $C_{a,j}$
Input: User-defined weight for overlapping field-polys: c_{ff} User-defined weight for overlapping gate-polys: c_{gg} User-defined weight for overlapping gate and field-polys: c_{gf} Origin x (left) coordinate $C_{a,j} = b$ Origin x (left) coordinate $C_{a-1,j} = i$ Width of cell $C_{a,j} = w_a$ Width of cell $C_{a-1,j} = w_{a-1}$
Output: Value of $HCost(a, b, a-1, i)$: horizontal cost of placing cell C_a at placement site b when C_{a-1} is placed at site i.
Algorithm: 01.**Case** $a = 1$: $HCost(1, b, 0, i) = 0$ 02.**Case** $a > 1$ **Do** 03. For every pair of left poly geometry in cell $C_{a,j}(LP)$ and right poly geometry in cell $C_{a-1,j}(RP)$ that overlap**{** 04. Call the geometries LP, RP /* Let Hspace be the spacing between LP and RP. Let AF_l be the largest assist correct spacing smaller than Hspace. Let the CD degradation slope(delta CD/delta spacing) for AF_l be $slope(l)$. */ 05. Split the vertical overlap between LP and RP into field-to-field O_{ff}, field-to-gate O_{fg} and gate-to-gate O_{gg} overlaps. /* Calculate overlap weight between RP and LP */ 06. $weight = slope(l) \times (Hspace - AF_l) \times (c_{ff}O_{ff} + c_{gf}O_{gf} + c_{gg}O_{gg})$ s.t. $AF_{l+1} > Hspace \geq AF_l$, 07. $HCost(a, b, a-1, i)$ += $weight$ **}**

Figure II.3: Horizontal Cost ($HCost$) calculation.

$C_{a,j}$ and let x_a denote its (leftmost) placement coordinate in the given standard-cell row, where coordinates increase from left to right. In addition, let $\delta_{a,j}$ denote the placement perturbation of cell $C_{a,j}$ to adjust the spacing between cells. $\delta_{a,j}$ is positive if the cell is moved towards the right and negative otherwise. Then the **assist-correct placement perturbation problem** is:

VCost(a,b) of Cell $C_{a,j}$
Input: $C_{a,j}$: a^{th} cell in j^{th} row User-defined weight for overlapping field-polys: c_{ff} Origin x (left) coordinate $C_{a,j} = b$
Output: $VCost(a, b)$: vertical cost of placing cell C_a at placement site b.
Algorithm: 01. **Case** $j = 1$: $VCost(a, b) = 0$ 02. **Case** $j > 1$ **Do** 03. For every pair of bottom poly geometry in cell $C_{a,j}(FB)$ and top poly geometry in cell $C_{h,j-1}(FT)$ that overlap**{** 04. Call the geometries FB, FT /* Let Vspace be the vertical spacing between FT and FB. Let AF_l be the largest assist correct spacing smaller than Vspace. Let O_{ff} denote the field-to-field overlap lengths. */ 05. $weight = slope(l) \times c_{ff}O_{ff} \times (Vspace - AF_l)$ s.t. $AF_{l+1} > Vspace \geq AF_l$, 06. $VCost(a, b) \mathrel{+}= weight$ **}**

Figure II.4: Vertical Cost ($VCost$) calculation.

$$
\begin{aligned}
&\text{Minimize} \sum \mid \delta_{a,j} \mid \\
&\delta_{a,j} + x_{a,j} - x_{a-1,j} - \delta_{a-1} - w_{a-1} + s_{a,j}^{LP^f} + s_{a-1,j}^{RP^g} \quad \in \quad AF \\
&\text{s.t. } LP^f \text{ and } RP^g \text{ overlap at horizontal cell row} \\
&s_{a,j}^{FB^m} + s_{h,j-1}^{FT^n} \quad \in \quad AF \\
&\text{s.t. } FB^m \text{ and } FT^n \text{ overlap at vertical cell row}
\end{aligned}
$$

The objective can be made aware of cells in critical paths by a weighting function. Since the available number of allowable spacings is very small, obtaining a completely assist-correct solution is usually not possible in a fixed cell row width context. Therefore, a more tractable objective is to minimize the expected CD error at a predetermined defocus level.This "continuous" version of the problem is similar in nature to placement legalization of row-based placements but with manufacturability-based cost metrics instead of traditional wirelength metrics. Placement legalization has been previously solved in literature [25] using dynamic programming techniques. We solve this

"continuous" version of the above problem with the following dynamic programming recurrence.

$$
\begin{aligned}
Cost(1,b) = \quad & | \; x_1 - b \; | \\
Cost(a,b) = \quad & \lambda(a) \; | \; (x_a - b) \; | + \\
& Min_{i=x_{a-1}-SRCH}^{x_{a-1}+SRCH} \{Cost(a-1,i) + \alpha HCost(a,b,a-1,i) + \beta VCost(a,b)\}
\end{aligned}
$$

$Cost(a,b)$ is the cost of placing cell a at placement site number b. The cells and the placement sites are indexed from left to right in the standard-cell row. α and β give the relative importance between $HCost$ and $VCost$. Typically, $HCost$ has more weight because $HCost$ is related to gate printability which determines device performance. $HCost$ is the measure of total expected CD degradation of vertical poly geometries at the worst defocus value for the cell. It can be thought of as the weighted change in area of vertical poly geometries in the cell. Similarly, $VCost$ is the measure of total expected CD degradation of horizontal poly geometries at the worst defocus value.

Note the above memory-less cost structure which ensures that once the optimal solution up to cell i is obtained, it contains the optimal solution up to cell $i-1$. This optimal substructure is essential for dynamic programming. We restrict the perturbation of any cell to $\pm SRCH$ placement sites from its initial location. This helps contain the delay and runtime overheads of AFCorr placement post-processing. λ is a factor which decides the relative importance of preserving the initial placement and the final AFCorr benefit achieved for each given cell instance; in the current implementation, λ is directly proportional to the number of critical timing paths that pass through the given cell instance. $HCost$ and $VCost$ corresponds to the printability deterioration under defocus conditions for the horizontal and vertical interactions respectively. $Cost(a,b)$ depends on the difference between the current nearest-neighbor spacing of the polys and the closest assist feature correct spacing. The methods that we use to compute $HCost$ and $VCost$ are shown in Figure II.3 and Figure II.4. $slope(l)$ is defined as delta CD difference over delta pitch between AF_l and AF_{l+1}. Thus, perturbation cost is a function of $slope$, length and weight of overlapped polys, and space for SRAF insertion. Our algorithm takes a legal placement as an input, and outputs a legal placement with better depth of focus properties. In addition, $VCost$ depends on the number of abutted cells, L and R,

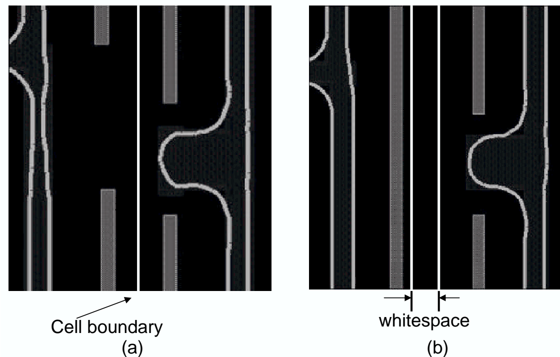


Figure II.5: (a) cell placement before horizontal AFCorr and (b) cell placement after horizontal AFCorr.

and the number of field-to-field-poly interactions. The runtime of the AFCorr algorithm is $O(ncell \times SRCH)$, where $ncell$ is the total number of cells in the design.

Figure II.5 shows an example of resist image profile with and without AFCorr technique. Horizontal-forbidden pitch is caused by interactions of poly geometries in the same row. After cell placement-perturbation in horizontal direction, additional SRAFs can be inserted at increased whitespace between cells and thus pattern printability is enhanced. In addition, vertical forbidden pitch violation is caused by inter-cell row interactions. As seen in Figure II.6 (a), there is not enough space between the vertically adjacent poly geometries (coming from cells in adjacent cell rows) which results in less SRAFs than needed. By moving the cell in upper row leftwards, this violation can be removed and printability enhanced.

II.A.4 Experimental Setup and Results

We synthesize the *aes* and $alu128$ benchmark design from *Opencores* in *Artisan TSMC 0.13µm* and *Artisan TSMC 0.09µm* libraries using *Synopsys Design Compiler v2003.06-SP1*. *aes* synthesizes to 12993 cells and 10286 cells in 130 nm and 90 nm technologies, respectively. $alu128$ synthesizes to 13279 cells and 8722 cells in 130 nm and 90 nm technologies, respectively. The synthesized netlists are placed with row utilization ranging from 50% to 90% using *Cadence First Encounter v3.3*. All designs are trial routed before running timing analysis. On the lithography side, we use *KLA-Tencor Prolith v9.1* to generate models for OPC. *Mentor Graphics Calibre v9.3_5.12* is used for

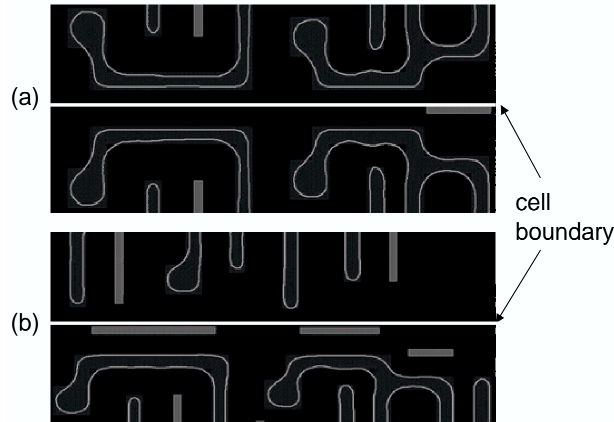


Figure II.6: (a) cell placement before vertical AFCorr and (b) cell placement after vertical AFCorr.

model-based OPC, SRAF OPC and optical rule checking (ORC). Photo simulation is performed with wavelength $\lambda = 248$nm and numerical aperture NA = 0.6 for 130 nm and $\lambda = 193$nm and NA = 0.75 for 90 nm. An annular aperture with $\sigma = 0.85/0.65$ is used for both processes.

We use three printability quality metrics. *Forbidden Pitch Count* is the number of border poly geometries estimated as having greater than 10% CD error through-focus. *EPE Count* is the number of edge fragments on border poly geometries having greater than 10% edge placement error at the worst defocus level. This is estimated by ORC. *SB Count* is the total number of scattering bars or SRAFs inserted in the design. A higher number of SRAFs indicates less through-focus variation and is hence desirable. We use $c_{fg} = c_{gg} = c_{ff} = 0.33$, $\lambda(a) = sitewidth/10 \times$ (number of top 200 critical paths passing through cell a) and $SRCH = 20$.

We first evaluate lithography printability of AFCorr with H- and V- assist correction. Proximity plot with fixed line width for the 0.13μm RET is illustrated in Figure II.7. CD degradation increases through-pitch as the defocus level increases. Patterns in the pitches of over $0.4\mu m$ before OPC are outside the allowable tolerance range at the worst defocus of 0.5μm. After bias OPC, pitches up to $0.38\mu m$ are allowable for CD tolerance while all pitches larger than 0.38μm should be forbidden. After evaluating SRAF OPC patterns with the worst defocus model, a set of forbidden pitches of

0.13μm technique is obtained: [0.37, 0.51), [0.635, 0.73), [0.82, 0.95), and [1.09, 1.17) (microns). Forbidden pitches still remain after SRAF OPC even though SRAF insertion considerably reduces forbidden pitches in comparison to bias OPC. On the other hand, proximity plot with SRAF OPC for 90 nm technology is illustrated in Figure II.8. Resist CDs after SRAF OPC are evaluated with the worst defocus model of 0.4μm. Resist CDs violate the allowable CD tolerance[6] as distance between SRAF and poly increases. A set of forbidden pitches of resist CD for 90 nm RET is calculated: [0.3, 0.41), [0.45, 0.57), [0.64, 0.73), and [0.78, 0.89) (microns). Generated SRAF rules may be summarized as shown in Table II.1. SRAF width and SRAF-to-pattern space are respectively 40nm and 120nm for 90 nm technology.

Table II.2 shows the results of horizontal and vertical forbidden pitches with various H vs. V weights. Increasing weight of HCost reduces the number of horizontal forbidden pitches while increasing the number of vertical forbidden pitches. H- and V-forbidden pitch counts are reduced by 94%-100% and 76%-100% for 130 nm, and by 96%-100% and 87%-100% for 90 nm, respectively. The design with 0.9 α for HCost and 0.1 β for VCost weights results in the highest reduction of total forbidden pitch counts and is chosen to evaluate SB count, running time, etc. Figure II.9 shows that the total number of SRAFs increases as the utilization decreases, due to increased whitespace between cells. The benefit of AFCorr decreases with lower utilization because the design already has enough whitespace for SRAF insertion. Due to the additional number of SRAFs inserted there is a small increase in SRAF OPC runtime ($< 3.6\%$) and final data volume ($< 3\%$). Reductions of EPE and forbidden pitch are investigated for each utilization as shown in Figure II.10. Total Forbidden Pitch Count is reduced by 89%-100% in 130 nm and 93%-100% in 90 nm. EPE Count is reduced by 80%-98% in 130 nm and 83%-100% in 90 nm. In addition, SB Count improves by 0.1%-7.4% for 130 nm and 0%-7.9% for 90 nm. Note that these numbers are small as they correspond to the entire layout rather than just the border poly geometries. The change in estimated post-trial route circuit delay ranges from -7% to +11%. All of these results for AFCorr are summarized in Table II.3.

[6] Allowable CD tolerance is assumed to be 10% of minimum line width in the worst defocus level.

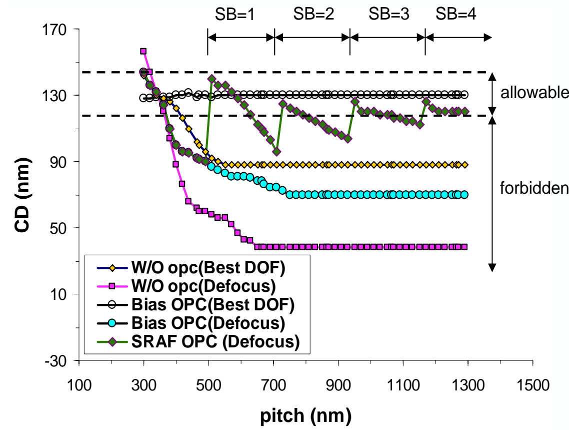


Figure II.7: Through-pitch proximity plots for 130 nm technology: best focus without OPC, worst defocus without OPC, worst defocus with BIAS OPC, and worst defocus with SRAF OPC are shown.

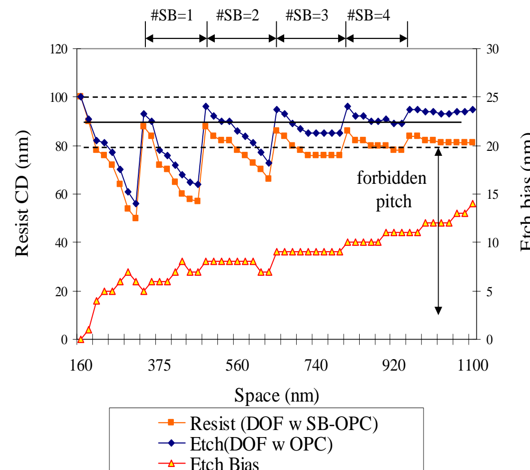


Figure II.8: Through-pitch proximity plots and etch skew for 90 nm technology: worst defocus with SRAF OPC and worst defocus with etch OPC (left Y-axis), and etch bias (right Y-axis) are shown.

Table II.1: SRAF rule table in 0.13μm and 0.09μm lithography.

	0.13μm Lithography		0.09μm Lithography	
	Pitch($X : \mu m$)	Slope	Pitch($X : \mu m$)	Slope
#SRAF = 0	$0 \leq X < 0.51$	0.28	$0 \leq X < 0.41$	0.162
#SRAF = 1	$0.51 \leq X < 0.73$	0.22	$0.41 \leq X < 0.57$	0.075
#SRAF = 2	$0.73 \leq X < 0.95$	0.105	$0.57 \leq X < 0.73$	0.062
#SRAF = 3	$0.95 \leq X < 1.17$	0.07	$0.73 \leq X < 0.89$	0.050
#SRAF = 4	$1.17 \leq X$	0.02	$0.89 \leq X$	0.012

Table II.2: Summary of Forbidden pitch results. Forbidden pitch counts slightly change based on different H- vs. V-weights.

	Utilization (%):	90		80		70		60		50	
	H:V weight	H F/P	V F/P	H F/P	V F/P	H F/P	V F/P	H F/P	V F/P	H F/P	V F/P
130 nm	0.9:0.1	4002	92	290	21	2	5	0	0	0	0
	0.7:0.3	5234	60	533	15	5	2	1	0	0	0
	0.5:0.5	5878	54	573	14	10	1	2	0	0	0
90 nm	0.9:0.1	4639	82	541	21	10	5	0	0	0	0
	0.7:0.3	5321	70	721	15	11	2	1	0	0	0
	0.5:0.5	6072	43	891	14	14	1	1	0	0	0

Table II.3: Summary of AFCorr results. Runtime denotes the runtime of SRAF and etch dummy insertion and model-based OPC. The AFCorr perturbation runtime ranges from 2 to 3 minutes for all test cases. GDS size is the post-SRAF OPC data volume.

	Utilization (%):	90		80		70		60		50	
	Flow:	Typical	AFCorr	Typical	AFCorr	Typical	AFCorr	Typical	AFCorr	Typical	AFCorr
130 nm	# Forbidden	20632	4094	3201	311	2011	7	1421	0	219	0
	# SB	158987	171691	173673	183860	185493	192578	195741	199704	212079	212412
	# EPE	4630	4721	5975	562	4276	15	1732	0	199	0
	Runtime (s)	7821	7902	7876	7934	7913	7973	7998	8013	8021	8121
	GDS (MB)	48.9	48.9	48.8	48.9	48.2	48.4	48.3	48.5	48.2	48.4
	Delay (ns)	4.2	4.6	4.5	4.7	4.5	4.6	4.6	4.9	5.2	5.4
90 nm	# Forbidden	22121	4721	4821	562	3812	15	2001	0	321	0
	# SB	115652	128387	139182	147520	153904	156244	164264	165649	182572	182666
	# EPE	7523	1262	4813	532	2131	107	1329	59	163	5
	Runtime (s)	6211	6327	6322	6431	6482	6499	6521	6571	6672	6692
	GDS (MB)	43.1	43.3	43.2	43.3	43.2	43.3	43.7	43.8	44.6	44.8
	Delay (ns)	2.7	2.7	2.6	2.6	2.4	2.47	2.8	2.9	3.1	3.2

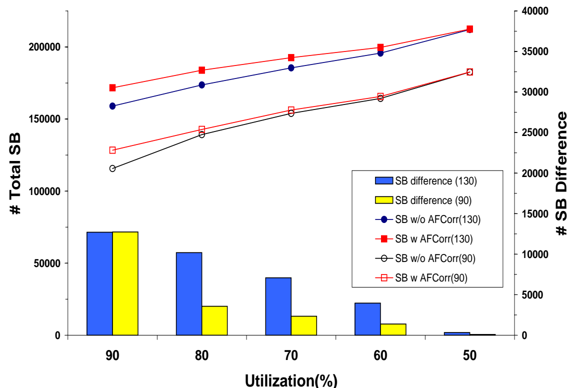


Figure II.9: Number of SRAFs with and without AFCorr for each of five different utilizations.

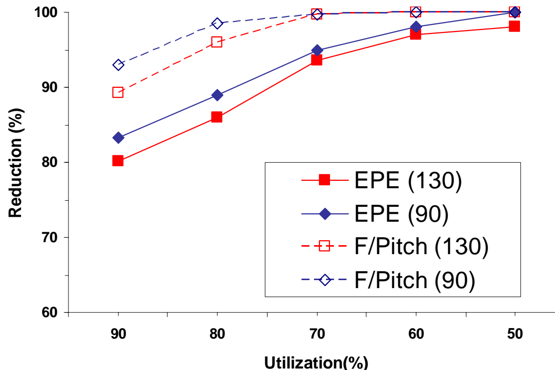


Figure II.10: Reductions of forbidden pitches with AFCorr methodology for each of five different utilizations.

II.B Etch Dummy Correctness

II.B.1 Etch Dummy and Layout Impact

Insertion of etch dummy features has been introduced to reduce the CD difference between resist and etch processes for 90 nm and below technology nodes. In dry etch processes such as plasma, ion, and reactive ion etch (RIE), different consumptions of etchants with different pattern density lead to etch skew between dense and isolated patterns. For example, all available etchants in areas with low density are consumed rapidly, and thus the etch rate then drops off significantly. In areas with high density of patterns, the etchants are not consumed as quickly. As a result, the proximity be-

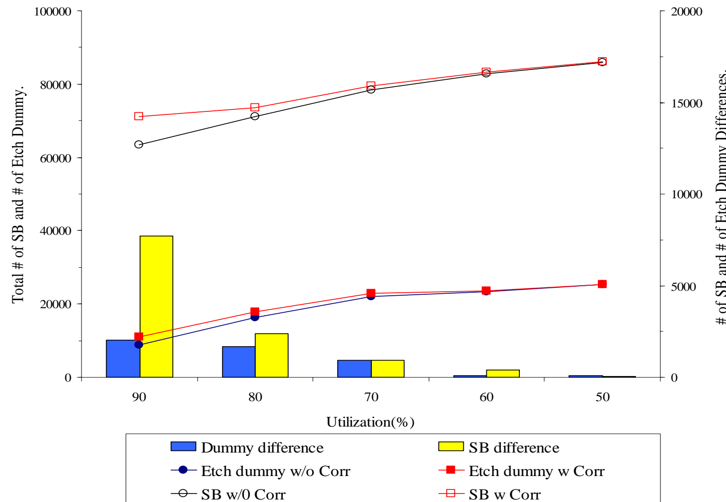


Figure II.11: Number of inserted SRAF and etch dummy features with various etch dummy insertion methodologies for each of five different utilizations.

havior of photo process differs from etch process as shown in Figure II.12. In general, the etch skew of two processes increases as pitch increases. When etch dummies are placed adjacent to primary patterns, a relatively isolated primary line will behave more like a dense line, and thus the etch dummies can reduce the etch skew. Moreover, the maximum relevant pitch is reduced through etch dummy insertion. This is an important consideration with respect to model-based OPC, which calculates the proximity effect of all patterns within a given proximity range, such that larger proximity range increases OPC runtime. Granik [4] observes that the proximity range of the etch process is around 3μm, which prevents conventional model-based OPC from delivering a good OPC mask within feasible turnaround time.

Etch Dummy Insertion Problem. Given a layout, find an etch dummy placement such that the following conditions are satisfied:

- Condition (1): Etch dummies are inserted between primary patterns with certain spacing to reduce etch skew between resist and etch processes.
- Condition (2): Etch dummies are placed outside of active-layer regions.

Thus, *Etch Dummy Correction Problem* is to determine perturbations to inter-cell spacings so as to insert optimal the number of etch dummy. Forbidden pitch correc-

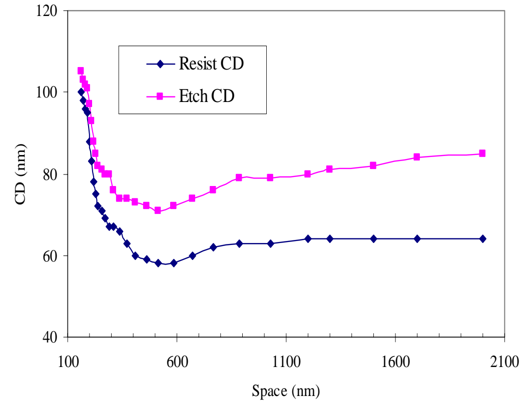


Figure II.12: Different proximity behaviors between photo and etching processes with pitch.

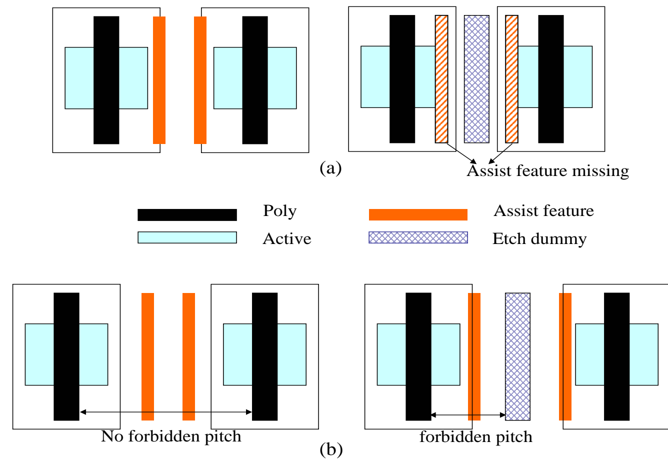


Figure II.13: Conflict between SRAF and etch dummy rules: (a) assist feature missing, (b) forbidden pitch occurrence.

tion in the resist process is required after inserting etch dummy because the etch dummy cannot be placed too close to primary patterns due to Condition (2). Etch dummy insertion can make printability of resist process worse in certain pattern configurations. Figure II.13 shows examples such as (a) assist features missing and (b) forbidden pitch occurrence. Assist features can be missed due to lack of space between primary pattern and etch dummy, even when there is enough space to insert multiple SRAFs before etch dummy insertion. New forbidden pitches for assist features can occur in the spacing between poly and etch dummy due to mismatch between rules for assist feature and etch

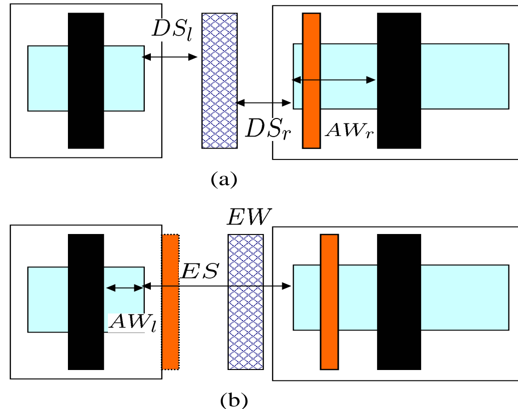


Figure II.14: (a) Typical etch dummy generation, (b) SRAF-aware etch dummy generation.

dummy corrections. Therefore, we now propose a new *Corr problem*: combination of assist feature and etch dummy insertion methods is as follows.

Assist feature and Etch Dummy Correction Problem. Given a standard-cell layout, determine perturbations to inter-cell spacings so as to simultaneously insert SRAFs in forbidden pitches and insert etch dummies to reduce etch skew.

II.B.2 SRAF-Aware Etch Dummy Generation

To reduce etch proximity, at most one etch dummy for each active (or diffusion) geometry is needed since the etch skew depends on pattern-to-pattern spacing regardless of local pattern density [11], i.e., etch skew decreases as the spacing is reduced. SRAFs and etch dummies have been generated by rule-based methods with look-up tables (LUTs) since simulation tools are much slower than rule-based tools. Typically, etch dummy rules consist of etch dummy-to-active space (*DAS*), etch dummy width (*EW*) and etch dummy-to-dummy space (*DDS*) with respective values of 120nm, 100 nm and 200nm being typical for 90 nm technology. Let ES denote the space between active geometry in the left and right cells as shown in Figure II.14. Let ED_1 and ED_2 denote the required spaces to insert one and two etch dummies in ES, respectively. For typical methods of etch dummy insertion, minimum space rules for one and two etch dummies are $ED_1 = 2 * DAS + EW$ and $ED_2 = 2 * DAS + 2 * EW + DDS$, respectively. The first

etch dummy in the typical etch dummy rule is always placed at the center of the space between two active geometries, while the active-to-etch dummy space for the second etch dummy is always according to the space rule, DAS.

Once etch dummies have been inserted for only etch proximity control, the spacing between poly and etch dummy may not be appropriate for SRAF insertion. Figure II.14(a) shows an example where the left-hand side SRAF cannot be inserted due to lack of poly-to-etch dummy spacing. Let AW_l and AW_r denote the distances between border polys and active geometries located at left- and right-cells, respectively. Let $AF = AF_1, \ldots, AF_m$ denote a set of "assist-correct" spacings. AF_j is the jth member of the set of assist feature correct spacings AF. Let AS_l and AS_r denote additional spacings needed for assist-correctness in the left- and right-cells, respectively. To avoid missing SRAFs and occurrence of forbidden pitches, we propose a new *SRAF-Aware Etch Dummy Method* (SAEDM) considering active width (AW) during insertion of etch dummy, as follows:

$$
\begin{array}{ll}
Minimize & \text{index values of } j \text{ and } k \text{ in a set } AF \\
s.t. & AS_l = AF_j - (AW_l + DAS) \text{ and } AS_r = AF_k - (AW_r + DAS), \\
 & \text{and } (AS_l + AS_r) \leq (ES - ED_1)
\end{array}
\qquad \text{(II.1)}
$$

SAEDM searches assist-correct spacing with minimum index values in a set AF, so that the sum of the additional spacings AS_l and AS_r corresponding to assist-correct spacings is less than $(ES - ED_1)$. Let DS_l and DS_r denote the left- and right-spaces from etch dummy to border active geometries in left- and right-cells, respectively. Thus, new etch dummy spaces of $DS_l = AS_l + DAS$ and $DS_r = AS_r + DAS$ are both assist-correct and etch dummy-correct. Note that the etch dummy after SAEDM is no longer located at the center of an active-to-active space since DS_l differs from DS_r, as shown in Figure II.14(b). Table II.4 compares DS_l and DS_r values returned by the typical etch dummy method and by SAEDM.

II.B.3 Corr Placement Algorithm

Assist-correct pitch rules are violated if there is not enough space to insert AS_l and AS_r. We now describe an etch-dummy correction *EtchCorr* placement perturbation

Table II.4: Comparison of etch dummy rules between conventional etch dummy method and SAEDM. Note that AS_l + AS_r = ES - ED_l.

Etch dummy rules		Typical method		SAEDM	
	ES (X)	DS_l	DS_r	DS_l	DS_r
#ED = 0	$0 \leq X < ED_1$				
#ED = 1	$ED_1 \leq X < ED_2$	$(ES\ \text{-}EW)/2$	$(ES\ \text{-}\ EW)/2$	$AS_l + DAS$	$AS_r + DAS$
#ED = 2	$X \leq ED_2$	DAS	DAS	$AS_l + DAS$	$AS_r + DAS$

algorithm using intelligent whitespace management. EtchCorr differs from AFCorr as follows: (1) EtchCorr is based on the active-to-cell outline spacing while AFCorr is poly-to-cell outline spacing. (2) EtchCorr calculates the virtual positions of etch dummy in order to both insert SRAF in assist-correct spacing and etch dummy in etch dummy-correct spacing. Let etch dummy-correct spacing (EDS) be inter-device spacing with etch skew less than 10% of minimum line width. Thus the etch dummy-correct perturbation problem is to minimize design perturbation to insert etch dummies optimally and thus to reduce etch skew between resist and etch processes. However, as we discussed, a new design correction technique *Corr* which combines two methods of assist-correct (AFCorr) and etch-correct (EtchCorr) placements is required to avoid conflict between assist feature and etch dummy insertions.

In the following, we describe the single-row Corr perturbation algorithm. Let $s_a^{RP_i}$ and $s_a^{RA_j}$ respectively denote the spacing between the right outline of the cell and the ith right border poly, and the spacing between the right outline of the cell and j^{th} active geometry. $s_a^{RE_i}$ is the spacing from right border poly to etch dummy as shown in Figure II.15. Let δ denote a cell placement perturbation to adjust the spacing between cells. ES, the space between border actives, is $x_a - x_{a-1} - w_{a-1} + s_{a-1}^{RA_i} + s_a^{LA_i}$. Then the **Corr placement perturbation problem** is:

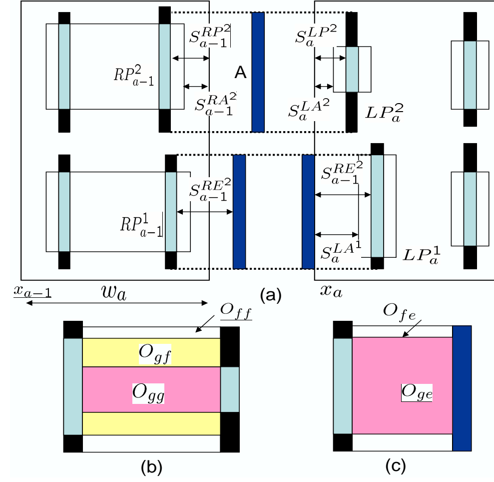


Figure II.15: The placement perturbation problem for assist and etch dummy insertion. (a) Multiple interactions of gate-to-dummy and field-to-dummy, (b) Overlap area when there is no etch dummy, (c) Overlap area in the presence of etch dummy.

$$
\text{Minimize } \sum \mid \delta_i \mid \text{ such that}
$$

$$
\begin{cases}
\text{If } (ES < ED_1) \\
\delta_a + x_a - x_{a-1} - \delta_{a-1} - w_{a-1} + s_{a-1}^{RP^i} + s_a^{LP^i} \in AF \\
\delta_a + x_a - x_{a-1} - \delta_{a-1} - w_{a-1} + s_{a-1}^{RA^i} + s_a^{LA^i} \in EDS \\
\text{s.t. } -SRCH \leq \delta_{a-1} \text{ and } \delta_a \leq SRCH \\
\text{otherwise} \\
S_{a-1}^{RP^i} - S_{a-1}^{RA^i} + S_{a-1}^{RE^i} + \delta_{a-1} \text{ and } S_a^{LP^i} - S_a^{LA^i} + S_a^{LE^i} + \delta_a \in AF \\
S_{a-1}^{RE^i} + \delta_{a-1} \text{ and } S_a^{LE^i} + \delta_a \in EDS \\
\text{s.t. } -SRCH \leq \delta_{a-1} \text{ and } \delta_a \leq SRCH
\end{cases}
$$

The terms $AFCost$ and $EDCost$ denote assist feature and etch dummy costs, respectively. $AFCost$ depends on the difference between the current nearest-neighbor spacing of the polys and the closest assist-correct spacing. The methods of computing $AFCost$ and $EDCost$ are shown in Figure II.16[7]. The formulation is similar to the AFCorr when the space between border actives is not enough for a dummy insertion. However, $Corr$ perturbation problem calculates poly-to-dummy spacings instead

[7]The Figure shows only H-AFCost computation for simplicity. We do not compute vertical EDCost as primary focus of etch dummies gate CD control

of poly-to-poly spacings when there are etch dummies between cells. O_{gg}, O_{ff} and O_{gf} respectively correspond to the length of overlap areas of gate-to-gate, field-to-field and gate-to-field-poly as shown in Figure II.15. O_{ge} and O_{fe} correspond to the overlapped length of gate-to-dummy and field-to-dummy. In addition, c_{gg}, c_{ff}, and c_{gf} are proportionality factors which specify the relative importance of printability for gate and field-poly. W_1 and W_2 are user-defined weights for $AFCost$ and $EDCost$, respectively.

$$
\begin{aligned}
Cost(1,b) = & \mid x_1 - b \mid \\
Cost(a,b) = & \lambda(a) \mid (x_a - b) \mid + \\
& Min_{i=x_{a-1}-SRCH}^{x_{a-1}+SRCH}\{Cost(a-1,i) \\
& +W_1 AFCost(a,b,a-1,i) + W_2 EDCost(a,b,a-1,i)\}
\end{aligned}
$$

II.C Modified Design and Evaluation Flow

To account for new geometric constraints that arise due to SRAF OPC in physical design, we add forbidden pitch extraction and post-placement optimization into the current ASIC design methodology. Figure II.17 shows the modified design and evaluation flows in the regime of forbidden pitch restrictions. Of course, we must assume that the library cells themselves have been laid out with awareness of forbidden pitches, and indeed our experiments with commercial libraries confirm that there are no forbidden pitch violations in poly geometries within individual commercial standard cells. (Our method solves forbidden pitch violations between placed cells.) SRAF insertion rules to enhance DOF margin are determined based on best and worst focus models.[8]

The post-placement optimization is performed based on forbidden pitches and slopes of CD error within them. After AFCorr (SAEDM and EtchCorr techniques will be described in detail below), we obtain a new placement which is more conducive to insertion of SRAFs, thus allowing a larger process window to be achieved. The two layouts generated by conventional and assist-correct flow undergo comprehensive SRAF OPC. The amount and impact of the applied RET is a function of the circuit layout.

[8] In general, the best focus is shifted from zero to about 0.1μm due to refraction in the resist. The worst defocus is the maximum allowable defocus corner for manufacturability in a lithography system. As the CD tolerance is +/-10%, the worst defocus model can be extracted by the Bossung plot in Fig II.1, i.e., worst defocus model is 0.5 μm for 130 nm technology.

Cost(a,b,a-1,i) of Cell C_a
Input: User-defined weights for poly-to-poly overlap: c_{gg}, c_{ff}, c_{gf} User-defined weights for poly-to-dummy overlap: c_{ge}, c_{fe} Width of cell $C_a = w_a$
Output: Value of $AFCost$ and $EDCost$: costs for corrections of assist feature and etch dummy of placing cell C_a at placement site b, respectively.

Algorithm:
/* Cost of placing cell C_a at placement site 'b' when cell C_{a-1} is placed at site 'i'. */
Let $AFspace$ denote the horizontal spacing between RP and LP.
Let ES denote the horizontal spacing between RA and LA.
Let $AFslope(j)$ be defined as ratio of resist CD degradation and change in pitch between AF_j and AF_{j+1}.
Let $EDslope(j)$ be defined as ratio of etch CD degradation and poly-to-dummy space.
Let ED_1 denote the required spaces to insert one etch dummy.
01. **Case** $a = 1$: $AFCost(1, b) = EDCost(1, b) = 0$
02. **Case** $a > 1$ **Do** {
03. **If** $(AFspace < ED_1)$ {
04. For every pair of left poly geometry in cell $C_a(LP)$ and right poly geometry in cell $C_{a-1}(RP)$ that overlap**{**
05. Call the geometries LP, RP
06. Split the vertical overlap between LP and RP into field-to-field O_{ff}, field-to-gate O_{fg} and gate-to-gate O_{gg} overlaps.
07. $AFweight = AFslope(j) \times (AFspace - AF_j)$ $\times(c_{ff}O_{ff} + c_{gf}O_{gf} + c_{gg}O_{gg})$ s.t. $AF_{j+1} > AFspace \geq AF_j$
08. $EDweight = EDslope(AFspace)\ \times(c_{ge}O_{ge} + c_{fe}O_{fe})$
}}
09. **Else** {
10. For every pair of pattern geometries in $C_a(LP)$, $C_{a-1}(RP)$ and dummy that overlap**{**
11. Call the geometries LP, RP, and a dummy pattern
12. Split the vertical overlap between poly and dummy into gate-to-dummy c_{ge} and poly-to-dummy c_{fe} overlaps.
13. $AFweight = AFslope(j) \times (AW_l + DS_l - AF_j) \times (c_{ge}O_{ge} + c_{fe}O_{fe})$
14. $AFweight+ = AFslope(l) \times (AW_r + DS_r - AF_l) \times (c_{ge}O_{ge} + c_{fe}O_{fe})$
15. $EDweight = (EDslope(AW_1 + DS_l) + EDslope(AW_r + DS_r)) \times (c_{ge}O_{ge} + c_{fe}O_{fe})$
}}
16. $AFCost(a, b, a-1, i)$ += $AFweight$
17. $EDCost(a, b, a-1, i)$ += $EDweight$
}

Figure II.16: The algorithm for $AFCost$ and $EDCost$ computations.

Thus we can evaluate how assist-correct placement impacts circuit performance and printability/manufacturability according to the metrics of SRAF insertions and edge placement errors (EPE). The following sections give more details of forbidden pitch extraction and design implementation.

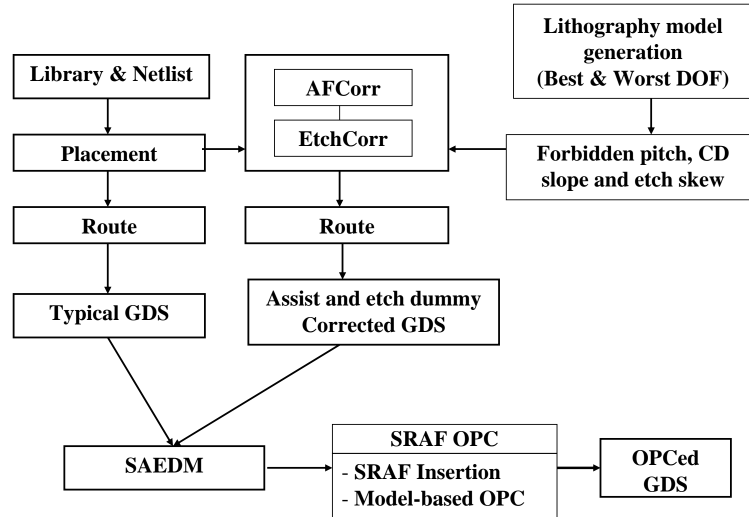


Figure II.17: The modified design and evaluation flows: Note the added steps of forbidden pitch extraction, SAEDM and post-placement optimization to ASIC design flow.

To account for new geometric constraints that arise due to SRAF and etch dummy in physical design, we extract forbidden pitch, CD slopes of resist and etch process with pitch, and CD skew induced by etch process. Post-placement optimization generates a new placement wherein the coordinates of cells have been adjusted to avoid the forbidden pitches and to reduce etch skew. The target etch process consists of three etch steps: 10 second breakthrough etch step to get through the BARC (Bottom Anti-Reflective Coating), 60 second main etch step, and 36 second overetch step. The breakthrough and main etch steps in the model produce a fair amount of deposition, taking the resist profile to 100 nm. The overetch step trims this back to the 90 nm range. A set of etch parameters is shown in the Table II.5. We only consider the first etch step to remove Si Nitride because second etch, step to etch gate-poly, does not impact CD variation with pitch [57]. Figure II.18 shows the calibrated vertical profile of dense patterns after resist and etch processes.

We use same benchmark designs as AFCorr and evaluate pattern printability with combinations of (1) SAEDM, (2) AFCORR + SAEDM and (3) AFCorr + EtchCorr + SAEDM. We generated SRAF rules with results in Table II.1. SRAF width and SRAF-to-pattern space are 40nm and 120nm, respectively. In addition, dummy-to-active space, etch dummy width and etch dummy-to-dummy space correspond to 120nm, 100 nm and

Table II.5: Etch process conditions for the simulator in 90 nm technique.

Stage	Etch time (sec)	Material	Vertical etch rate (sec)	Horizontal etch rate (sec)	Faceting Parameter Parameter
1	10	ArF Sumitomo	10.66	-0.6	0.5
		AZ BarLi-2	10.52	-0.7	0.0
		Si Nitride	10.28	-0.7	0.0
2	60	ArF Sumitomo	0.3	-0.12	0.5
		AZ BarLi-2	3.4	-0.2	0.0
		Si Nitride	30.4	-0.3	0.0
3	36	ArF Sumitomo	10.65	0.9	0.5
		AZ BarLi-2	0.25	1.0	0.0
		Si Nitride	0.0	1.5	0.0

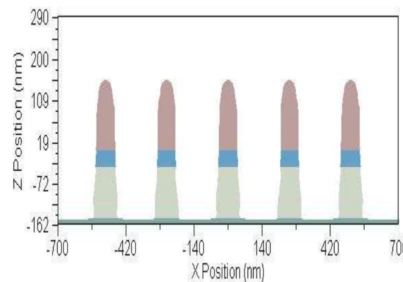


Figure II.18: Calibrated vertical profile after photo and etch processes.

200nm respectively. However, the spacing between active and etch dummy is varying because SAEDM changes the space with the active width. Resist and etch CDs vary with location of the SRAF insertion, and resist CDs violate the allowable CD tolerance as distance between SRAF and poly increases. The trend of etch CD follows the variation of resist CD. The skew of resist and etch CDs continuously increases with pitch and is not saturated by 1.1μm as shown in Figure II.8.

After Corr placement perturbation, we obtain a new placement wherein the coordinates of cells minimize the occurrence of forbidden pitches of resist and etch processes. Total cost of Corr is calculated using specific weights of resist and etch costs (in the results reported, we use respective weights $W_1 = 0.9$ and $W_2 = 0.1$). Note that our post-placement perturbation problem reduces to the previously-studied AFCorr problem if $W_2 = 0$.

We evaluate the reduction of Forbidden Pitch Count with various etch dummy insertion methodologies in resist and etch processes shown in Table II.6. After (1)

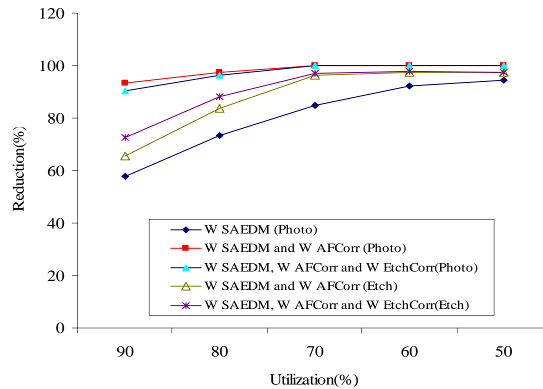


Figure II.19: Reductions of forbidden pitches with various etch dummy insertion methodologies for each of five different utilizations.

SAEDM, Forbidden Pitch Count of photo process can be reduced by 57% - 94% with various utilizations because etch dummy-to-poly spacings become assist-correct. However, Forbidden Pitch Count of the etch process may increase by up to 6% in certain layout configurations since the SAEDM increases the poly-to-etch dummy spacing. The Forbidden Pitch Counts of etch process in (2) SEADM+AFCorr and (3) Corr+SAEDM are respectively reduced by up to 64%-97% and 73%-98% across a range of utilizations as shown in Figure II.19. (3) Corr+SAEDM facilitates additional SRAF and dummy insertion by up to 10.8% and 18.6%, respectively. Figure II.11 shows that the total number of SRAFs and etch dummies increases as the utilization decreases. Note that these numbers are small as they correspond to the entire layout rather than just the border poly geometries. EPE Count is reduced by 91%-100% in resist process and 72%-98% in etch process. The change in estimated post-trial route circuit delay ranges from 3% to 5.8%. The increases of data size and OPC running time overheads of Corr are within 3% and 4% respectively. The runtime of Corr placement perturbation is negligible ($\sim$ 5 minutes) compared to the running time of OPC ($\sim$ 2.5 hours). All of these results for Corr are summarized in Table II.7.

Table II.6: Forbidden pitch results with various etch dummy insertion methodologies in resist and etch processes.

	Utilization (%):	90	80	70	60	50
Photo	W/O SAEDM, W/O AFCorr, W/O EtchCorr	37433	31314	29216	26765	21282
	W SAEDM, W/O AFCorr, and W/O EtchCorr	15743	8330	4423	2075	1198
	W SAEDM, W AFCorr, and W/O EtchCorr	2432	822	23	0	0
	W SAEDM, W AFCorr, and W EtchCorr	3566	1116	51	0	0
Etch	W/O SAEDM, W/O AFCorr, and W/O EtchCorr	15816	8812	4656	4345	3530
	W SAEDM, W/O AFCorr, and W/O EtchCorr	16418	9729	5282	5002	4209
	W SAEDM, W AFCorr, and W/O EtchCorr	5423	2221	172	109	92
	W SAEDM, W AFCorr, and W EtchCorr	4321	1032	143	92	92

II.D Conclusions and Ongoing Work

In this work, we have presented novel methods to optimize etch dummy insertion rules and detailed standard-cell placements for improved etch dummy and assist feature insertion. We obtain a practical and effective approach to achieve assist feature compatibility in physical layouts. The *AFCorr*, as an approach to achieve assist feature compatibility, leads to reduced CD variation and enhanced DOF margin. We also introduce a dynamic programming-based technique, *Corr*, to achieve etch dummy insertion correctness in the detailed placement step of standard-cell based chip implementation. Corr with SAEDM leads to reduced CD variation and increased insertion of assist features and etch dummies. For our test industrial cases we have observed the following.

- In lithographic printability evaluation of AFCorr, H- and V-forbidden pitch counts for border poly geometries are reduced by 94%-100% and 76%-100% for 130 nm, and by 96%-100% and 87%-100% for 90 nm, respectively. For EPE count, the reductions of 80%-98% in 130 nm and 83%-100% in 90 nm are obtained. We also achieve up to 7.6% increase in the number of inserted scattering bars.

- In pattern printability evaluation, the forbidden pitch count of photo process between polysilicon shapes of neighboring cells is reduced by up to 54%-94%, 92%-100%, and 90%-100% for SAEDM, SAEDM+AFCorr and SAEDM+Corr, respectively. The forbidden pitch count of etch process of SEADM+AFCorr, and SAEDM+Corr is respectively reduced by up to 64%-97% and 73%-98% with across a range of utilization. Corr with SAEDM facilitates additional SRAF and dummy insertion by up to 10.8% and 18.6%, respectively.

Table II.7: Summary of SAEDM+Corr results. Runtime denotes the runtime of SRAF and etch dummy insertion, as well as model-based OPC. The Corr perturbation runtime ranges from 4 to 5 minutes for all test cases. GDS size is the post-OPC data volume.

	Utilization(%):	90		80		70		60		50	
	Flow:	Typical	SAEDM+Corr	Typical	SAEDM+Corr	Typical	SAEDM+Corr	Typical	SAEDM+Corr	Typical	SAEDM+Corr
Photo	# EPE	42102	3723	32434	1243	29349	98	28721	13	23134	2
	# Forbidden	37433	3566	31314	1116	29216	51	26765	0	21282	0
	# SB	63349	71051	71101	73501	78513	79432	82820	83230	85991	86026
Etch	# EPE	17209	4812	9213	1200	4820	182	4821	109	3890	109
	# Forbidden	15816	4321	8812	1032	4656	143	4345	92	3530	92
	# Dummy	8876	10911	16240	17920	22088	23001	23390	23499	25237	25309
Other	Runtime (s)	6835	7011	7451	7535	7529	7632	7685	7698	7943	7944
	GDS (MB)	41.1	42.3	41.2	43.2	42.2	42.3	42.9	42.8	43.6	43.6
	Delay (ns)	2.478	2.305	2.458	2.602	2.522	2.47	2.867	3.176	3.113	3.046

- In impact on other design metrics, the increases of data size, OPC running time and maximum delay overheads of Corr are within 3% and 4%, respectively. In addition, maximum delay overhead of 6% is within noise of the P&R tools [5]. The runtime of Corr placement perturbation is negligible ($\sim$ 5 minutes) compared to the running time of OPC ($\sim$ 2.5 hours).

- *Restricted Design Rules (RDRs).* It may be possible to derive forbidden pitches from a set of restricted design rules which allow only few pitches in the layout. With increasing adoption of RDRs, "legalization" of layouts with respect to these rules becomes an important task where an AFCorr like methodology can be useful. Part of our ongoing work analyzes "correct-by-construction" standard-cell layouts which are always EtchCorrect in any placement scenario. We intend to compare such an approach with EtchCorr placement perturbation in terms of design as well as manufacturability metrics.

- *Extension to Other Layers.* Placement affects shapes on diffusion, contact, metal1 and metal2 layers besides polysilicon layer. The Corr cost function can be extended to include forbidden pitches from these other layers. In future technology generations, process window for local metal layers is becoming a big concern and again since placement determines most of local metal layout, AFCorr like technique can help.

- *Preferential Treatment of Devices.* Certain devices and cells may be able to tolerate more process variation than others in the design. For instance, narrow devices typically have smaller process window. We are investigating techniques to bias the AFCorr and EtchCorr solution in favor of such devices to reduce timing and power impact and increase overall parametric yield.

- *Leakage Power Objective.* We currently use a linear function of CD error as the objective function in the Corr algorithms. With leakage being a dominant concern with scaling geometries, a somewhat exponential function of CD error may be more appropriate.

II.E Acknowledgments

Chapter II is in part a reprint of "Detailed Placement for Enhanced Control of Resist and Etch CDs", to appear in *IEEE Transactions on CAD*. I would like to thank my coauthors Chul-Hong Park and Dr. Andrew B. Kahng.

III

Dealing with Systematic Focus-Dependent FEOL CD Variation

The ability to more accurately model and manage variability of designs in ultra-deep sub-micron technology has become ever more critical in the success of technologies beyond 90 nm CMOS process. Since critical dimensions are scaling faster than our ability to control them, e.g., effective gate-length of a transistor, variability has become an increasingly more important design issue [47, 118]. It is recognized that traditional static timing approach is becoming too conservative to predict the actual performance of a design [49, 47, 50, 51]. Progress has been made to employ statistical techniques to model variability of circuit performance. A general probabilistic framework has been proposed to improve the accuracy of timing prediction [49]. Several approaches to address the correlations due to path re-convergence and proximity gates are studied [47, 50, 52].

Across Chip Linewidth Variation (ACLV) is a major contributor to timing variation in ultra-deep sub-micron technology. Other sources of variation includes metal thickness, temperature, voltage, oxide thickness, etc. Here we focus on the systematic components of ACLV for the polysilicon level. There are many sources which contribute to ACLV: through-pitch variation, through-process variation, topography variation, mask variation, etching, etc. Due to the complex interaction between these sources

of variation, ACLV has been modeled as a random phenomena [51]. In reality, at least 50% of ACLV is systematic [54, 53]. The systematic through-pitch variation is the major contributor to variation at nominal process condition, and the systematic through-focus variation is the major contributor for through process condition. These systematic variations can be modeled very accurately once a physical layout is completed.

There have been a number of papers studying pattern-dependent variability. In particular, [29, 44] examined the characterization and impact of systematic spatial gate-length variation on the performance of circuits. The authors claim that the systematic spatial intra-chip CD variability, rather than pattern-dependent proximity effects, are the primary cause of circuit delay variation and speed degradation. They classified all gates into 18 different categories based on the orientation (vertical or horizontal) and spacing between neighboring poly lines (i.e., dense, denso, and isolated) to capture the interaction between the optical lithography process and local layout patterns. Dense is defined as minimum spacing, denso represents an intermediate distance, and all others are labeled as isolated. *Lgate* values are then measured from the testchips to build *Lgate* maps. The *Lgate* maps containing spatial information of each gate are fed into a tool to generate modified netlists depending on the location of gates. Results show that about 17% of critical path delay variation and up to 25% of timing error and performance loss occur without proper consideration of systematic spatial gate-length variation components.

This chapter proposes a method for timing analysis and optimization methods to compensate for pattern-dependent variation.

III.A Systematic Variation Aware Timing Analysis

Static timing analysis based on worst-case timing is a common sign-off process adopted in ASIC. In reality, the worst-case timing is never achieved in actual hardware. One reason is because the worst-case timing approach assumes ACLV of transistors is independent, which is never the case. This is addressed by various statistical timing approaches [47, 52, 49]. The other reason is because worst-case timing model does not consider the systematic components of ACLV which can be predicted accurately based on physical context of the gates. In this section, we investigate the systematic

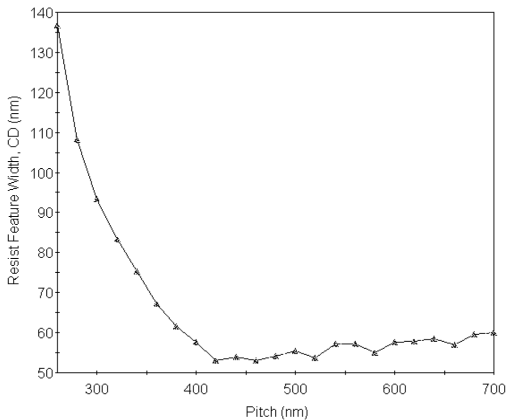


Figure III.1: An example of through-pitch variation for an annular illumination system with λ=193nm and NA=0.7 calculated using *Prolith* [57]. The drawn dimension is 130 nm. Notice the "radius of influence" of less than 600nm.

components of ACLV, its magnitude and timing impact. We propose a systematic-variation aware timing methodology. We show that by taking into consideration the systematic variations, we can reduce the best-case to worst-case timing spread by up to 40% in a traditional static timing analysis. Similar impact of this type of systematic aware modeling in statistical timing analysis can also be expected but it is not covered in this section.

III.A.1 Systematic Variation: Magnitude and Impact

Through-pitch linewidth variation at nominal process condition, can best be demonstrated by a typical plot of linewidth versus pitch such as Figure III.1 . The plot shows printed linewidth systematically increases as the pitch increases. Optical proximity correction (OPC) is a technique used to correct this systematic effect. OPC is a necessary VLSI mask data processing step in today's technology [36]. It attempts to correct the distortion of printed image due to proximity environment of the designed shapes at nominal process condition.

While the correction reproduces the intended design shapes on wafer as best as possible, it is not perfect. We demonstrate even with OPC, there is systematic linewidth variation at nominal process condition. This is done by applying standard OPC to

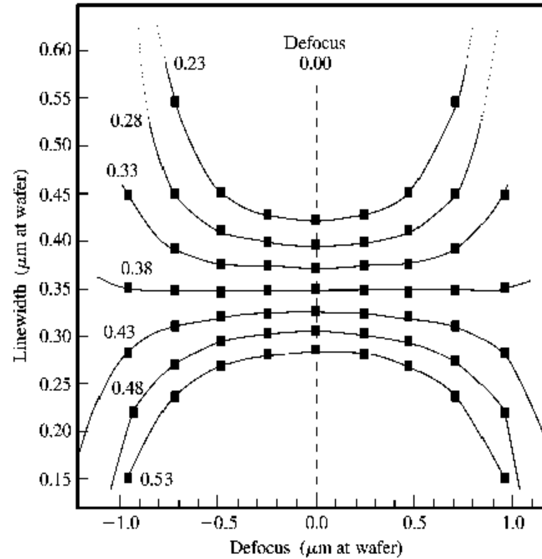


Figure III.2: Linewidth vs. defocus for 0.35 μm width with varying spacing for a given exposure setting [55]. Notice the systematically different behavior or isolated and nested lines.

parallel poly lines with different spacing, and then measure the average linewidth of the simulated wafer image of the corrected poly lines. Our results indicate a systematic decrease in linewidth as the pitch increases from 300nm to 600nm. The magnitude of the variation is about 10% of the target linewidth. This implies that the nominal timing model can have as much as 10% discrepancy from the target linewidth, assuming delay varies linearly with gate-length.

Through-focus linewidth variation is illustrated by a standard Bossung plot (e.g. Figure III.2) of linewidth vs. defocus condition. For a binary mask technology, the Bossung plot depicts opposite behavior of dense lines and isolated lines. For a dense line, the linewidth increases as the process goes out of focus, the “smiling” part of the plot. For an isolated line, the linewidth decreases as the process goes out of focus, the “frowning” part of the plot. This systematic effect is somewhat mitigated by insertion of assist features [56] but never completely. The through-focus variation can account for up to 30% of the total ACLV budget.

III.A.2 Overview of the Systematic Variation Aware Static Timing Methodology

Timing model for a standard-cell is characterized by a very intensive simulation process. It is reduced to a set of formulas which predict delay of input to output paths based on parameters such as gate-length, temperature, voltage, oxide thickness, etc. The corners of the model assume the worst case condition for each parameter. In particular, worst case gate-length is assumed to be the maximum possible gate-length variation. In reality, as described above, gate-length variation can be predicted more accurately based on the spatial environment of each gate. The accurate prediction will remove at least half of the best-case to worst-case spread of the gate-length. In this subsection, we describe a timing methodology which takes into consideration the systematic variation of gate-length. We also quantify the pessimism caused by using the worst case assumption.

Accounting for pattern-dependent Variation

Traditional timing methodology assumes perfect printing of the gates and hence computes timing of a design based on the target gate-length. Model-based OPC tries to achieve the target gate-length but is never able to correct the design perfectly. The reasons may include OPC-unfriendly layout patterns and limitations of the OPC algorithm as well as constraints on runtime. As a result there always is some iso-dense bias in printing of polysilicon shapes. Isolated lines tend to print smaller (or larger depending on the process) than nested or dense shapes. This pitch-dependent variation of printed gate-length is systematic and hence can be predicted. After placement, spacing between all gate shapes is known and hence printed shapes can be predicted accurately.

OPC can be performed on the layout and lithography simulations can be done to predict the printed shape on final wafer. The critical dimension or gate-length can then be measured from this simulated print-image of the layout for each device. This more accurate gate-length can then be used to predict the timing of the device, cell and hence the entire design more accurately. The problems with such an elaborate approach are as follows.

- *OPC is computation intensive.* Model-based OPC is very computation intensive.

Typical numbers range from about 1100 seconds for a small 5900 gate design (see Table III.1) to several CPU days for modern multi-million gate designs. Moreover, image simulation of the entire design is also very time consuming and hence not suitable for use during the design process which may involve many SP&R iterations.

- *Library characterization is an involved process.* Characterizing a standard cell for continuously varying gate-lengths (or *Critical Dimension, CD*) of all the devices within it is a herculean task if not an impossible one. Performing circuit-level timing on the entire design with accurate gate-lengths is also not feasible due to runtime and scalability constraints.

Our method for accounting through-pitch variation in static timing has three major components namely: accurate CD measurement, constructing timing libraries and contextual timing analysis. We describe these parts of our flow next.

CD measurement To circumvent the problems of full-chip OPC and elaborate characterization, we adopt a library based OPC approach similar to one described in [154]. Individual library cells are corrected conservatively in a typical placement environment. The placement environment is emulated using a set of dummy geometries. For example, see Figure III.3. Further details can be found in [154]. The average gate-length[1] is then measured for all devices in the gate. These "printed" gate-lengths are then used to predict timing for the devices.

This library-based OPC approach is accurate enough because the radius of influence for 193nm steppers is about 600nm. I.e., features beyond 600nm of any given device have negligible impact on its printing. As a result, the devices which are not at the periphery of the cell have an environment which is almost identical to their actual placement environment. Therefore, the CD predicted for them after library-based OPC is very close to the CD predicted for them after full-chip OPC.

Devices which lie at the boundary of the cell are not as accurately predictable by the library-OPC approach. For these devices, we use a through-pitch CD simulation approach. We construct a look-up table which matches pitch to printed CD for the

[1]The gate-length varies along the width of the device. We do a simple averaging of the CD. We believe this to be a reasonable approximation as device delay varies almost linearly with gate-length.

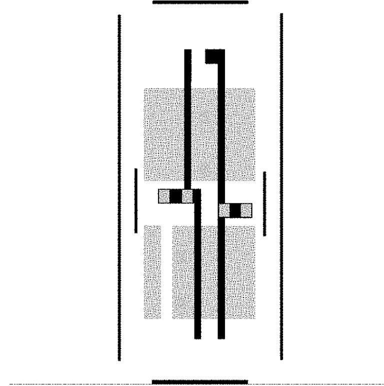


Figure III.3: A library-based OPC environment setup for a simple NAND gate. Note the dummy poly geometries inserted to emulate the impact of neighboring cells on the cell under consideration.

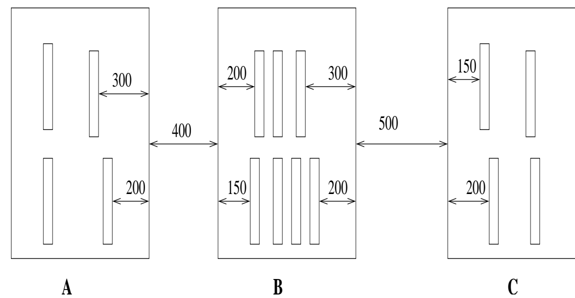


Figure III.4: An example placement of cells A, B and C. For cell B, $nps_B^{LT} = 900, nps_B^{RT} = 950, nps_B^{LB} = 750, nps_B^{RB} = 900$.

given process. The CD measurements are again done post-OPC. The empirical model is constructed for a number of spacings up to 600nm. The placement of the cell in layout determines the CD to be used for these border devices. An example is shown in Figure III.4.

Constructing Timing Libraries In a placement, a cell's environment will depend on the neighboring cells (left and right in a horizontal cell placement row)[2] and the whitespace between the cell and its neighbors.

In a placement for a cell C_i, its environment is described by a set of four spacings nps_i^{LT} (distance of the device on the "left-top" to the nearest poly feature on the left in

[2]We do not consider "vertical" neighbors as they have negligible impact on gate CD.

the neighboring cell), nps_i^{RB} (distance of the device on the "right-bottom" to the nearest poly feature on the right), nps_i^{LB} and nps_i^{RT}.[3] These four space parameters enable us to determine the printed CD for the border poly features in the cell in the placement context using the through-pitch CD simulation results. Since continuous variation of these parameters makes a library difficult to characterize, we use *three* different values each of these parameters. This gives rise to 81 different versions of the same cells.[4]

For our current experiments, we assume delay of any timing arc from an input pin to an output pin in a cell to be linearly proportional to the gate-lengths of the devices involved in the transition. The devices involved are fixed for the worst-case transition. Though we use this linear approximation for simplicity, more accurate circuit simulation based analysis is also feasible. We construct timing look up tables (with varying load capacitance and input slews) for these 81 versions of the library cell master. As a result, we obtain a .lib which has 81 versions of each cell in the original library.

In-Context Timing Analysis After the library generation, the next step is to identify correct canonical environment for every cell instance in the layout and perform a contextual static timing analysis. We define four parameters for a cell C_i: s_i^{LT} (the distance of cell outline from the closest device on the "left-top" corner of the cell), s_i^{LB} (spacing between left-bottom device and the cell outline), s_i^{RT} and s_i^{RB}. Analyzing the placement (i.e., whitespace around the cell and the four s parameters for the given cell and its immediate neighbors) puts the given cell in the given layout into one of the 81 categories.

After annotating each cell instance with its correct version, we run static timing analysis with the expanded library. The result of this timing analysis takes into account iso-dense effects and the resulting through-pitch variation at the nominal focus and exposure.

[3]Note that the top and bottom spacings can be different as they correspond to p and n devices respectively which may not be aligned in the cell layout.

[4]81 is arrived as a compromise between accuracy and ease of implementation.

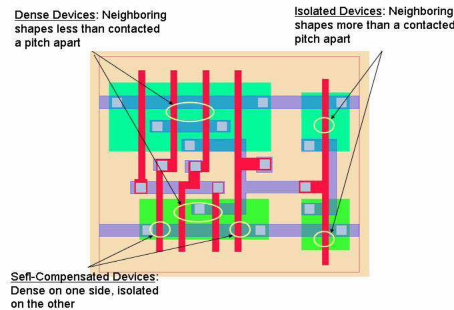


Figure III.5: Mixture of dense, isolated and self-compensated devices.

Taming Focus Variation

The next systematic component of variation that we account for in our proposed timing analysis methodology is the CD variation arising out of focus variation. Isolated and dense lines behave differently with defocus. Isolated lines tend to get thinner with defocus while dense lines get thicker. As a result, isolated devices get faster with focus variation while dense devices tend to get slower than nominal.

An important component of "process" corner for timing is gate length variation. A very important component of gate-length variation is focus variation. The systematic "smile-frown" behavior of focus-based variation of CD implies that depending on whether a certain timing arc involves isolated devices or dense ones, the worst-casing in one of its corners can be reduced. Moreover, there is some "self-compensation" of focus variation for timing arcs which involve both isolated and dense devices. See Figure III.5. As before, we analyze the devices in the layout and label them as isolated, dense or self-compensated depending on the spacing to the nearest poly line on the left and the right.[5] Next we label each timing arc (input pin to output pin transition) as "smile", "frown" or "self-compensated" depending on whether the devices involved in the transition are isolated, dense or self-compensated.[6]

[5]We assume "dense" spacing to be less than the contacted-pitch and anything larger to be "isolated". We ignore the "self-compensated" spacing for our current experiments as the number of possible spacings in the library layouts is limited.

[6]For purpose of this work, we assume the majority determines the nature. For example, if a timing arc involves two isolated and one dense device, then it is labeled as frowning. Better focus-sensitivity based characterization is possible but we limit ourselves for want of an accurate defocus print-image simulator.

We assume given a certain percentage contribution of focus variation to CD variation. For smiling timing arcs, we trim of that portion from the best-case gate-length. For frowning timing arcs, the worst-case gate-length is reduced while for self-compensated timing arcs worst-case as well as best-case gate lengths are impacted. As a result, timing uncertainty arising out of focus variation is reduced for *all* timing arcs in the design.

Computing the Corners

Traditional corner-based timing analysis uses slow, nominal and fast corners for process. The systematic variation aware static timing analysis flow proposed in this work reduces the pessimism and uncertainty caused by these variations.

To compute the impact of through-pitch variation, we draw test layouts consisting of parallel poly lines with fixed width and length but varying spacing. These test layouts are then corrected with the standard OPC flow and CD is measured to construct the look-up table described in subsection III.A.2. Denote the total range of CD variation after OPC by $\pm lvar_{pitch}$. We calculate (similarly defined) $\pm lvar_{focus}$ using the FEM (Focus Exposure Matrix) curves built from fabrication of test structures. We measure the CD variation with defocus (focus variation range is taken to be $\pm$300nm) for a number of pitches (ranging from minimum pitch to a pitch slightly larger than the contacted pitch). These variations are shown in the artificial Bossung plot in Figure III.6.

Let l^{nom} and l^{nom}_{new} denote the traditional nominal gate-length (independent of the cell layout and placement) and the iso-dense aware gate length respectively. Define l^{WC}_{pitch} and l^{BC}_{pitch} to be the worst-case and best-case gate-lengths after accounting for through-pitch variation in CD. Similarly, l^{WC} and l^{BC} be the corresponding numbers in the conventional flow. Then

$$\begin{aligned} l^{WC}_{pitch} &= l^{nom}_{new} + (l^{WC} - l^{nom} - lvar_{pitch}) \\ l^{BC}_{pitch} &= l^{nom}_{new} - (l^{nom} - l^{BC} - lvar_{pitch}) \end{aligned} \tag{III.1}$$

There are many factors affecting the best and worst case gate length, we are removing the variation due to pitch. In reality, there are dependency between the pitch

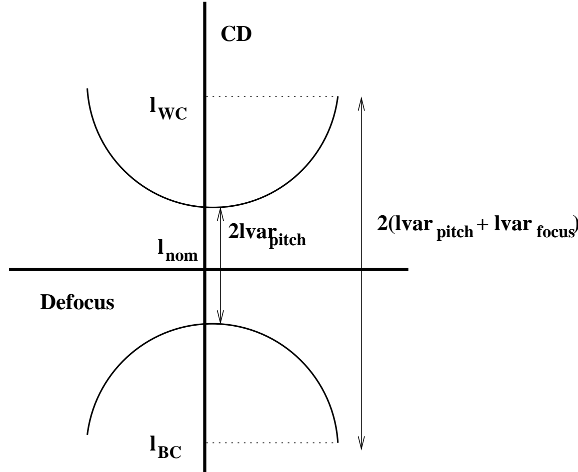


Figure III.6: An artificial Bossung curve at some given nominal exposure. The smile denotes the "most dense" feature in the technology while the frown denotes the "most isolated" one. It should be clear that the total span of CD variation ($= 2(lvar_{pitch} + lvar_{focus})$) is too pessimistic.

and the non-pitch factors. For the purpose of quantifying the potential impact of taking into consideration the systematic variation, this is a very good first order assumption. We will discuss what can be done to improve the accuracy in an actual systematic variation aware timing methodology later in this section.

Focus variation does not effect the nominal process corner. Moreover it may effect worst-case and best-case corners differently depending on whether the timing arc under consideration is smiling, frowning or self-compensated.[7] For smiling timing arcs, the values are

$$l_{smile}^{WC} = l_{pitch}^{WC} \tag{III.2}$$
$$l_{smile}^{BC} = l_{pitch}^{BC} + lvar_{focus}$$

Here, we are removing the variation due to focus from the best case, since it is not a factor for dense lines. Similarly, for frowning timing arcs,

$$l_{frown}^{WC} = l_{pitch}^{WC} - lvar_{focus} \tag{III.3}$$
$$l_{frown}^{BC} = l_{pitch}^{BC}$$

[7] In this work we do not consider "degree" of compensation for the lack of supporting data.

For self-compensated arcs, both worst-case and best-case timing is modified.

$$l_{selfcomp}^{WC} = l_{pitch}^{WC} - lvar_{focus} \tag{III.4}$$

$$l_{selfcomp}^{BC} = l_{pitch}^{BC} + lvar_{focus} \tag{III.5}$$

III.A.3 Experiments and Results

To quantify the magnitude of the pessimism of traditional STA, we take 10 most frequency used cells in a 90 nm standard-cell library, synthesize ISCAS85 benchmark circuits with the 10 cells, and then timed the synthesized and placed circuits for best-case, nominal ad worst-case. The corner case libraries are constructed with just the process corners while the voltage and temperature are kept the same across all the libraries. We do this to evaluate the benefit of the proposed timing methodology independent of any orthogonal effects.

We apply OPC to these 10 cell masters as described in earlier, using *Mentor Graphics Calibre*. Model-based OPC is performed using IBM 90 nm pre-production process models. To verify that through-pitch variation is sizeable even after model-based OPC, we measure CDs of full-chip standard model-based OPC and compare it with nominal gate-length. The distribution of error is given for an example circuit in Figure III.7. We see up to 20% variation in printed gate-length even after model-based OPC.

To evaluate effectiveness of the library-based OPC approach we compare the printed CD of library-based OPC with traditional full-chip OPC approach. The results are given in Table III.1. The table shows that about 50% of all devices corrected in a library-based OPC fashion fall within 1% error while nearly all devices have a printed gate-length within ± 6% of full-chip OPC. Moreover, most of the error-prone devices are likely to lie on the periphery of the cell which are accounted for in a "rule-based" fashion in our timing methodology.

We perform in-context timing analysis for the synthesized and placed circuits with the in-context timing model described earlier, by substituting the correct version of the timing model for each cell based on its placement. We generating the 81 versions of each cell with values of nps^{LT}, nps^{RT}, nps^{LB} and nps^{RB} each being put into one of the

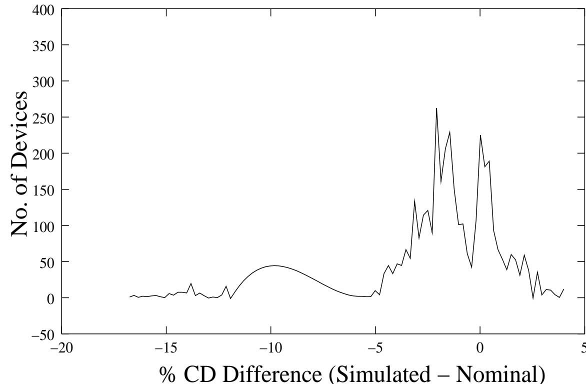


Figure III.7: Distribution of error for model-based OPC for c3540 ISCAS85 benchmark.

Table III.1: Comparison of library-based OPC and full-chip OPC. N-i% denotes % of devices with less than i% error compared to full-chip OPC. library OPC Runtime is 90 seconds for 10 masters.

Testcase	N-1%	N-3%	N-6%	Runtime (s)
C1355	58	83	97	477
C2670	45	78	96	747
C3540	40	77	96	1131
C432	35	76	97	185
C499	54	79	96	495

three bins: {400-500nm, 500-600nm, $\geq$ 600nm}. Since the radius of influence of 193nm steppers is about 600nm, any spacing larger than 600nm is isolated spacing and prints almost the same as a 600nm spacing. Since dense geometries print larger in the process, we use lower of the bin extremes (e.g., 400nm for 400-500nm bin) to be pessimistic in our timing estimates.

We compare the best-case, nominal and worst-case timing with the standard timing as described above. Assuming $lvar_{focus}$ and $lvar_{pitch}$ each to be 30% of the total gate-length variation [53], the results of systematic-variation aware STA are shown in Table III.2. Our results show that the best-case to worst-case timing spread is reduced by 28% to 40% in the systematic variation aware approach. Since majority of the devices in the layout are isolated (due to the whitespace distribution or the cell layout itself), the nominal timing improves when through-pitch variation is accounted for.

Table III.2: Comparison of traditional worst-case timing with systematic variation aware timing methodology. Nom, BC, WC denote nominal, best-case and worst-case corners of the library respectively.

Testcase	#Gates	Traditional Timing (ns)			New "Accurate" Timing (ns)			% Reduction in Uncertainty
		Nom	BC	WC	Nom	BC	WC	
C1355	2058	2.15	1.57	2.88	2.15	1.70	2.62	29
C2670	3655	5.07	3.74	6.64	5.05	4.04	5.96	33
C3540	5903	6.32	4.72	8.34	6.26	5.20	7.35	40
C432	968	5.77	4.21	7.70	5.70	4.53	6.88	32
C499	1728	2.30	1.66	3.10	2.29	1.79	2.82	28

III.A.4 Practical Systematic Variation Aware Timing

Our experiment demonstrates that there is substantial pessimism in the traditional static timing analysis by not considering the systematic components of ACLV. In this subsection, we propose a practical systematic variation aware timing methodology.

In order to produce more accurate in-context timing model for each standard cell, each cell will need to be "corrected" by the OPC process before it is characterized. This can be done by the library based OPC methodology proposed in [154], in which, gates in the cell are corrected by standard OPC processed on a per cell definition basis as opposed to be corrected in a per instance basis. Gates on the boundary can have several versions of correction based on context. In such an OPC methodology, the timing characterization of a cell can be performed based on the actual wafer image of the corrected gates in the cell.

Furthermore, we need to develop a parameterized gate-length model for each gate on the cell boundary. The model will predict the actual gate-length and its variation based on the proximity spatial information, i.e.,distance of the neighboring gate. From our earlier discussion, the nominal gate-length can be predicted by through pitch gate-length simulation, and the through-focus gate length variation can be predicted by a Focus Exposure Matrix (FEM) plot.

A timing model which includes the proximity spatial information as a parameter for input to output path delay will need to be constructed. More specifically, the input to output delay is parameterized by s_i^{LT}, s_i^{LB}, s_i^{RT}, s_i^{RB} as described previously. One naive way to construct such a model will be to perform extensive input to output delay path

simulation for each value of the boundary gate-length. A more efficient construction of such a model is a topic which will require separate investigation.

With such a timing model parameterized by proximity spatial information, the systematic variation aware static timing analysis can be performed after placement.

III.A.5 Conclusions and Ongoing Work

We have proposed a novel static timing methodology which accounts for systematic variation arising due to proximity effects and focus variation. The methodology brings process and design closer and has elements of RET, library characterization as well as conventional static timing analysis. We quantify the magnitude of the pessimism of traditional static timing analysis which neglects systematic components of ACLV. This can amount to as much as 40% tightening of the best-case to worst-case timing spread. In practice, ASIC hardware always performs better than traditional STA predicts. Even though, different compensating mechanisms has been built into traditional STA, e.g., IBM EinsTimer [47], systematic variation could be one key component which contributes to the discrepancy as suggested by our results.

We are refining our experiment for process technology which includes other RET such as Sub-Resolution Assist Features. We also plan to further quantify such pessimism by using statistical timing methodology with more realistic gate length distribution based on iso-dense attributes and proximity spatial information, as opposed to the simplistic Gaussian distribution of gate-length variation. Another process phenomenon not accounted for in our current experiments is exposure dose variation. Exposure variation can alter the nature of devices (i.e.,dense or isolated).

Our current work also investigates the implications of exposure variation on the proposed timing methodology. Systematic nature of focus-dependent CD variation suggests potential implications for compensating for such focus variation. At the design-level, isolated and dense timing arcs may be balanced within a critical paths to reduce the sensitivity of circuit delay to focus variation. This is discussed in the next section of the chapter.

III.B Self-Compensating Design for Focus Variation

Systematic variation can be mitigated to some extent by performing OPC and inserting assist features, but cannot completely be eliminated due to various reasons (modeling errors, algorithmic inaccuracies, process variations, etc.). The remaining linewidth variation due to layout is significant even after the use of complex RET techniques, with isolated and dense lines retaining opposite behavior under varying defocus [104]. Thus, there is a possibility of compensating for systematic variation in the design itself. This compensation can be achieved in two ways: 1) by ensuring that each standard cell is robust against focus variation, and 2) by intelligently constructing a robust circuit out of inherently non-robust building blocks or cells.

Self-compensated Cell Layout By self-compensated cell layout, we refer to a correct-by-construction methodology that relies on within-cell compensation of CD variation caused by focus variation. For example, variation can be compensated in series-connected NMOS, if one device becomes shorter (thus, faster) under defocus, and the other device becomes longer (thus, slower). This can be achieved by making one device "iso" and the other device "dense". The other way of generating self-compensated cells is to find spacing ranges in which the linewidth variation is negligible by focus variation. Each spacing between adjacent poly lines should be one of these values. In this work, we generate all the self-compensated cells by requiring poly spacing to be in the compensated spacing range (to be discussed further later in this chapter). We also explore the possibility of *single pitched-cells* where all poly spacings are set to one highly manufacturable value to eliminate the focus-dependent CD variation inside cells.

Self-compensated Physical Design This refers to compensation across cells (e.g., along a critical path). Consider two cells G1 and G2 that lie on the critical path $G1 \rightarrow G2$. Focus variation, if not corrected by applying expensive RETs, can cause variation in critical path delay and lead to potential timing failures or parametric yield loss. However, if G1 is explicitly made "iso" while G2 is made to act "dense", focus-dependent CD variation can be compensated. Assuming that iso and dense versions of library cells are available, designs that are robust to focus variation become possible.

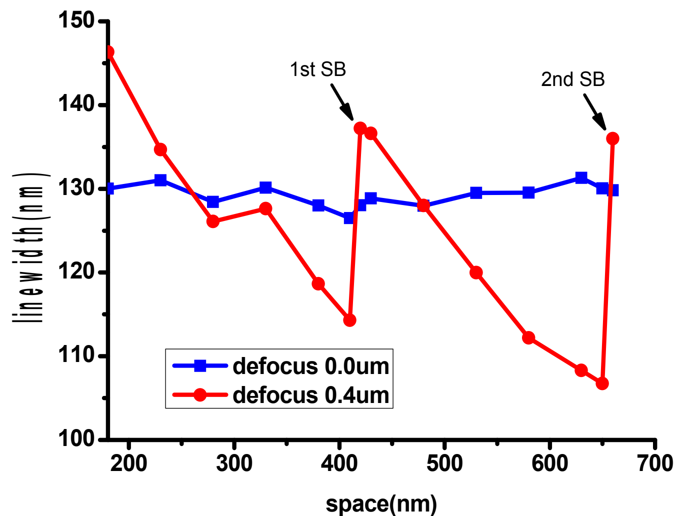


Figure III.8: Linewidth variation with spacing (SBs are inserted at 420nm and 660nm).

We compare and contrast the two approaches described above. For example, we seek to compare the area overheads of self-compensated libraries vs. across-cell optimizations. We generate each cell variant based on lithography simulation and the area overhead is then determined using place & route (PR) step. We also propose an integer linear programming (ILP) formulation to ensure timing is met across the expected focus range, and these results also allow us to determine the non-optimality of the heuristic approach.

III.B.1 Layout Generation

The work in [43] is based on the lithographic simulation results after OPC and SRAF insertion using Calibre WorkBench (WB) [162]. Critical dimensions at every space and focus level are obtained from the five-line patterns. However, no layouts are actually generated for iso, dense, and self-compensated cell variants. In that case, the area of each cell and its parasitics were estimated based on the deviation from the original layout spacings to new (iso/dense/self-compensated) spacings. To obtain better estimates of delay and area after placement and routing, we generate each version of cells using an automated layout generation tool [41].

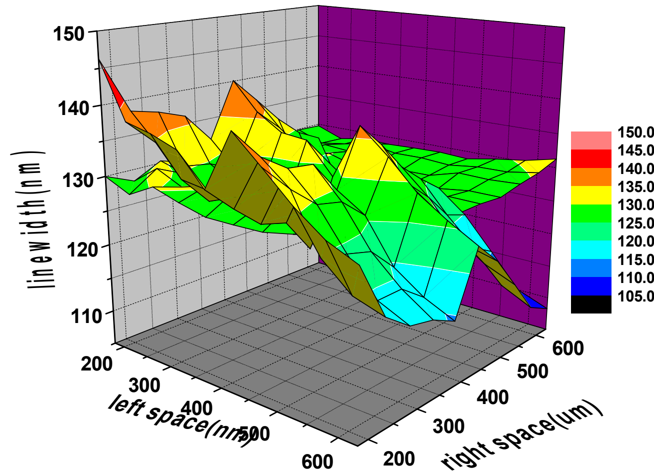


Figure III.9: Linewidth variation with asymmetric spacing for two defocus values, 0.0um and 0.4um. The nearly flat surface represents 0.0um defocus.

Table III.3: Parameters used in CalibreWB.

Parameters	Values
λ (wavelength)	248nm
NA (Numerical Aperture)	0.7
Illumination type	Annular
Scattering bar width	60nm
Scattering bar placement	180nm
Linewidth (nominal)	130 nm

Lithography Simulation. Lithography parameters used in Calibre WB [162] are shown in Table III.3. We use an optical lithography process with 248nm wavelength and numerical aperture of 0.7. Optical models are generated at 5 different defocus levels (e.g., 0.0, 0.1, 0.2, 0.3, 0.4um) and a constant threshold resist model is used. We assume that the optical characteristics are symmetric in defocus (i.e., +0.1um = -0.1um defocus).

CD Measurement. To find a specific spacing range for iso, dense, self-compensated, and single-pitched cells we perform lithography simulation after OPC and SRAF insertion at two defocus values; namely best and worst defocus values. The resulting

printed linewidths are then measured. *Leff* variations at the worst defocus value (0.4um) are used to construct criteria for spacing range of each variant of cells (i.e., iso, dense, self-compensated, and single pitched cells). The linewidth variation with spacing from 180nm to 660nm at 0.0um and 0.4um defocus level is shown in Figure III.8. The 3D graph with different left and right spacing from 180nm to 630nm with 50nm step is shown in Figure III.9. As can be seen from these graphs, due to the use of scattering bars, linewidth does not vary much at best focus even if the spacing between poly lines increase. The tolerance of the self-compensated devices is set at 4nm since the 3σ for the gate CD control is 4nm in 130 nm technology [2]. The 1st scattering bar is inserted at the spacing of 420nm, and the 2nd scattering bar is inserted at when spacing becomes 660nm. Therefore, we define allowable spacings for *dense* devices as 180nm (minimum space), 420nm (1st scattering bar point), and 660nm (2nd scattering bar point). We define 380nm-410nm and 600nm-650nm as the *iso* spacing range, and 260nm-320nm and 460nm-480nm as the *self-compensated* spacing regions. Finally, we select 480nm as the spacing value for *single-pitched* cells from the self-compensated regions, because the minimum spacing for contact is 420nm. Table III.4 summarizes the spacing criteria for cell generation. We can set our intended spacing of poly gates within technology files (ProTech [41]) to make each desired version of the library cells. A lumped-C model of capacitance is extracted and added into netlists to obtain more exact timing; to this end, we use a commercial parasitic extraction tool (CalibrePEX [162]).

To analyze iso/dense/self-compensated behavior with defocus, we use a five-line pattern and sweep the spacing between the three center lines from 180nm to 480nm. Scattering bar insertion and OPC are performed on these patterns using Calibre [162]. The average linewidth of the center line is then measured for each pattern. Figure III.10 shows the variation in this critical dimension for different spacing values at nine different defocus values. The figure shows distinct space ranges where the patterns behave as iso, dense or self-compensated.

Based on Figure III.9 and Figure III.10, we generate a look-up table (LUT) using the function CD = f(LS, RS, F), where LS is the left space, RS is the right space, and F is defocus. This allows us to obtain the exact degree to which specific patterns act isolated, dense, or self-compensated, and also to predict CD given defocus and spacings.

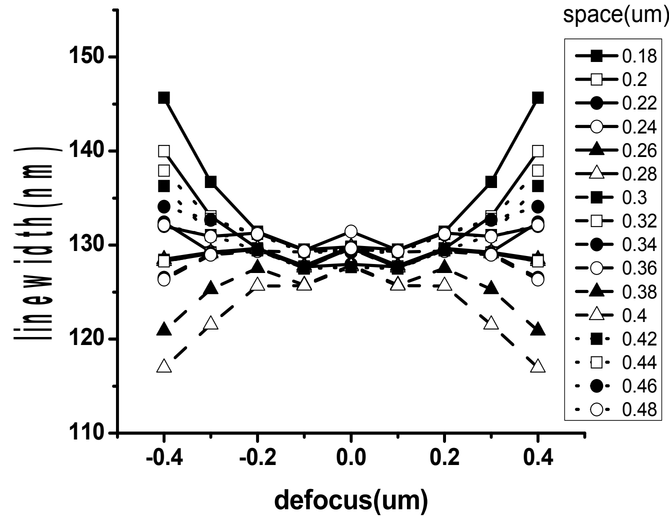


Figure III.10: Linewidth variation with defocus level (nominal linewidth = 130 nm).

Table III.4: Spacing criteria for cell generation (SB = Scattering Bar).

Cell version	Spacing range (nm)
Dense	180(min.), 420(1sb), 660(2sb)
Iso	380 ˜ 410, 600 ˜ 650
Self-compensated	260 ˜ 320, 460 ˜ 480
Single-pitched	480

The tolerance of the self-compensated devices is set at 4nm. Thus, if linewidths are 4nm larger than nominal at 0.4um defocus, we assume those patterns are “dense”; similarly, if linewidths are 4nm smaller than nominal, we classify the patterns as “iso”. Finally, if the CD variation is less than 4nm at 0.4um defocus, we consider the pattern “self-compensated”. The first scattering bar insertion point is at a spacing of 420nm, therefore, the “most-iso” pattern has a spacing of roughly 400nm. At 420nm spacing and above, the pattern reverts to “dense” behavior as a result of scattering bar insertion. At the “most-dense” spacing (180nm on each side), the linewidth increases 13% from nominal and in the “most-iso” case (i.e., 400nm on each side), the linewidth decreases 11% from nominal at the 0.4μm defocus point.

The optimal scattering bar placement and width depend on numerous factors such as wavelength (λ), numerical aperture (NA), illumination type, and others [33].

Reference [34] provides equations for optimal size and placement (defined as SRAF to main pattern spacing) of scattering bars, which are $(0.2 - 0.25) \times (\frac{\lambda}{NA})$ and $(0.55 - 0.75) \times (\frac{\lambda}{NA})$, respectively.

Edge Devices Special consideration is required for the edge devices. Edge devices are those devices that are closest to the standard-cell boundary. For example, since there is only one poly line for NMOS and PMOS in an INVX1 (minimum sized inverter) layout, these are all edge devices. We identify two different types of edge devices: Case 1 has no neighboring devices on either side (as in INVX1), while Case 2 has no neighboring device on exactly one side (e.g., left-most or right-most devices in cells except INVX1, INVX2 which have no fingers). To investigate the edge effect in Case 1, we first sweep the spacing from 180nm to 1μm symmetrically on both sides. Figure III.12 shows linewidth vs. spacing in Case 1. As can be seen from the graph, linewidth is insensitive to focus after two SRAFs are inserted on each side of poly line. For Case 2 edge devices, we fix one side at 180nm for dense and 380nm for iso devices. The spacing on the other side is swept up to 2μm. Figure III.11 shows the Case 2 edge effect of dense and iso cells respectively. When two adjacent poly lines are 1.2μm apart (i.e., 2 SRAFs are inserted at each side), the linewidth does not vary much even if the spacing becomes larger. Since the distance from edge devices to the cell boundary for all cells is over 600nm in this technology (making the distance of two neighboring poly lines more than 1.2μm), we assume that all edge devices of dense and iso cells in Case 2 follow the behavior seen in Figure III.11. For Case 2 edge devices of self-compensated and single pitched cells, we use the LUT linewidth value at one space from each layout and the other space at 660nm (2 SBs).

Library Construction The spacing between each poly line can be divided into 3 different ranges based on lithography simulation results. Specific space values are used to generate each layout variant of the cells. A layout synthesis tool is used to create the actual layouts in which all the spacings between poly lines are fixed to the values of each category. From the range of self-compensated spacing, one spacing value for which *Leff* variation is negligible is selected for single-pitched cells.

We consider 21 frequently used cells. All 5 variants of each cell are generated

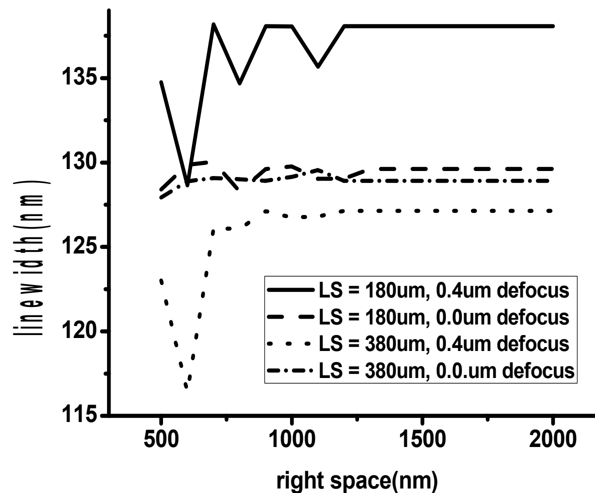


Figure III.11: Linewidth with spacing from 0.5μm to 2μm at 0.0μm and 0.4μm defocus in Case 2 for dense and iso cells.

Table III.5: Normalized area overhead of each cell version.

Cell version	**Actual Layout**	**Estimated**
Original	1.00	1.00
Dense	1.02	1.04
Iso	1.22	1.20
Self-compensated	1.10	1.10
Single-pitch	1.35	NA

using the same layout synthesis tool, namely original, dense, iso, self-compensated, and single-pitch. The original version is generated without any constraints in spacing, enabling the smallest area possible. The single-pitch version allows only one fixed spacing value between all poly lines. This single spacing/pitch value is chosen based on its insensitivity to focus variation. The cell height is set to 4.2μm. Table III.5 shows the average area overhead comparison found using both the actual cell layouts and the estimated areas taken from [43]. As can be seen from the table, the two approaches show similar values with self-compensated cell variants exhibiting ˜10% area increase on average. Also, single-pitch cells yield substantially higher area penalties than the other variants.

The LUT constructed from Figure III.10 gives CD of each variant of cells at two defocus values. Parasitic capacitances from the lumped-C model of parasitic extraction

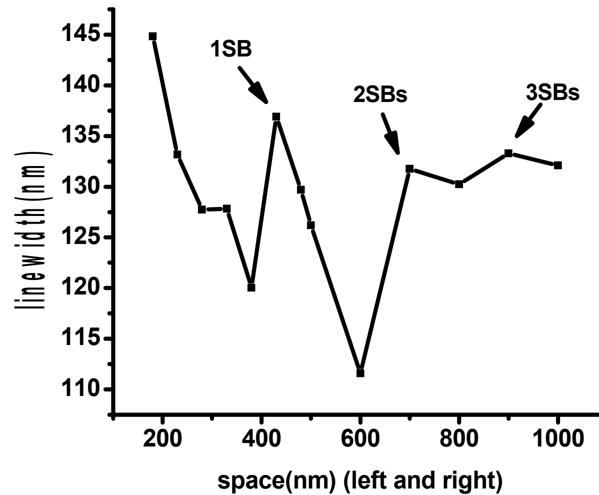


Figure III.12: Linewidth variation at 0.4μm defocus in Case 1. Arrows indicate scattering bar (SB) insertion points.

tool are included in netlists and timing and power characterization tool from Synopsys (Star-MTB) [46] is then extensively run to generate libraries of timing and power for each layout version of cells. The library (*.lib*) is standard Synopsys format which contains table-based timing and dynamic power and leakages information.

III.B.2 Self-Compensated Design Baselines

Self-compensated Cell-Based Design The most straightforward approach to creating a design that is insensitive to defocus in the lithographic process is to make each standard-cell element self-compensating in isolation. Based on the previous subsection we have created a self-compensated library and resulting circuit performance (area, delay, power) using this library will be used as a baseline in the results subsection.

Single-pitch Cell Design Designing circuits using a single pitch on the critical layer holds promise since a manufacturing process can be highly tuned to maximize manufacturability at a given pitch at the expense of printability of other pitches. We select one spacing value within the range of self-compensated spacings to generate single-pitch cells. Again, the resulting circuit performance using this library will be compared against

when evaluating the optimization approaches introduced below.

III.B.3 Optimization (Self-Compensated Physical Design)

As can be seen from the previous subsection, a more robust design with respect to focus variation is possible by using either self-compensated cells or single-pitched cells. Another option is to generate optimized circuits using both dense and iso cells to meet timing at all focus points. Optimization with a mix of dense and iso cells is possible both in the timing (i.e., to meet critical delay) and power (i.e.,to meet worst-case leakage constraints) domains. In this subsection we describe both heuristic and Mixed-Integer Linear Programming (MILP) solution methods to the self-compensated physical design problem.

Area-driven Timing Optimization The first optimization seeks to balance timing and area. We can generate new circuits that meet timing requirements through all defocus values by using both dense and iso cell variants; the goal will be to use as few iso variants as possible, thereby minimizing the area penalty.

Heuristic

Iso-dense self-compensating physical design can be viewed as a sizing problem. Since dense cells are slower (at worst-case focus) and smaller while iso cells are faster and bigger, we start with the circuit initially synthesized with dense cells, then swap in iso versions to meet timing at the worst-case defocus level.

Initially, synthesis with the dense library results in the slowest timing at worst defocus conditions with minimal area. The optimization of delay versus area is implemented using a sensitivity-based approach to minimize area penalty while instantiating iso counterparts of dense cells in the circuit to meet timing constraints. In our experiments, the required time at the primary outputs is set to be the worst-case delay with the original library at 0μm defocus. The sensitivity of all gates with respect to a change from "dense" to "iso" variants can be defined as in [38]:

$$Sensitivity = \frac{1}{\Delta A + K_1} \sum_{arcs} \frac{\Delta D}{slack_{arc} - S_{\min} + K_2} \qquad \text{(III.6)}$$

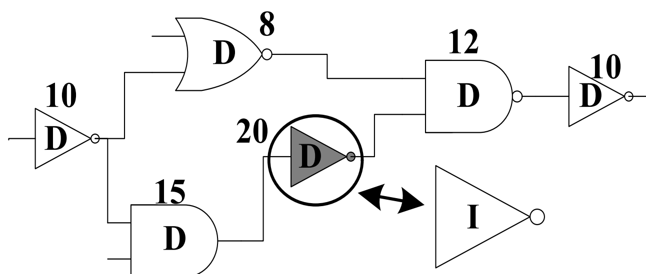


Figure III.13: Illustration of optimization process (D denotes "dense" and I denotes "iso" cell; numbers are example sensitivities of gates to swapping to "iso" counterparts).

Optimization
Input: focus Output: optimized circuits

While *worst_slack* is negative {
  Calculate sensitivities of all gates in the circuits
  Sort Sensitivities in non-increasing order
  Swap the "dense" version with "iso" cell, based on order of sensitivities
  Calculate *new_delay* of circuit
  Update *worst_slack*
}

Post-Processing

If *worst_slack* at intermediate defocus value is not negative
  Finish optimization and exit
Else
  Find the maximum-delay defocus point
  Perform optimization at the maximum-delay defocus point

Figure III.14: Optimization algorithm for self-compensated design.

Where A is the change in area and D is the change in delay due to the swap. *Smin* is the worst slack in the circuit when synthesized using the "dense" library, and the *arcs* consist of all rise and fall transitions from each input to output of the gate. The term *slackarc* is the difference between arrival and required times of the timing arc, and *K1* and *K2* are small positive numbers to ensure numerical stability of the expression. Pseudocode for our optimization process is shown in Figure III.14.

As the pseudocode indicates, we first sort sensitivities in non-increasing order. The gate with maximum sensitivity is then swapped with its corresponding iso version.

Incremental timing analysis updates the *worst_slack* value and new sensitivities are then calculated if the timing is not met. Figure III.13 shows an example of the swapping. The numbers in the illustration represent the sensitivity of each gate when changing from dense to iso counterparts. Since all gates are dense at first, the design may not meet timing at worst-case defocus. Changing from dense to iso compensates for defocus along critical paths. The algorithm iterates until timing constraints are met. Table III.6 shows the (absolute) increases in area and leakage power when dense cells are swapped with their iso counterparts. Inverters show negligible impact on cell area when exchanged with iso counterparts since there is often space in these cell layouts to make changes without impacting cell width. However, swapping more complex cells, such as NAND3 and NOR3, results in moderate area penalties. Most NAND2 or NOR gates, however, show relatively small leakage power increments compared to inverters (e.g., INVX12) since leakage in these cell types is inherently smaller. Absolute numbers are given since the algorithm directly operates on these and they will help shed light on decisions that the algorithm makes in the results subsection.

Even after the above optimization procedure (which ensures timing correctness at both best and worst focus conditions, assuming it is feasible), the circuit may not meet timing constraints at intermediate values of focus. This may occur since the optimization only uses information at perfect focus and worst-case defocus in guiding decisions, leading to potential timing failures when focus is non-linear or non-monotone. Thus, the timing constraint should be checked across defocus levels. We sweep the optimized circuits over all defocus values to find the maximum-delay focus condition. If the maximum-delay defocus point is out of the permissible focus range or the maximum delay is less than the required time, no further steps are needed. However, if the maximum-delay defocus point is within the permissible focus range, a post-processing step as described above is required to globally meet the timing constraint. At the maximum-delay defocus point, we can apply the same sensitivity-based optimization process shown above to ensure that the optimized circuit meets timing throughout the expected defocus range. Delay at intermediate focus values (e.g., 0.11μm, 0.37μm, etc.) is calculated by interpolation from pre-characterized cell delays at a small set of focus values (e.g., 0.1μm, 0.2μm, 0.3μm, etc.). In the interpolation, we assume CD to be a quadratic function of focus.

Table III.6: Area and leakage power change when dense cells are exchanged with iso Counterparts.

Cell	Area(μm^2)	Leakage(nW)
invx1	0	0
invx2	0	0
invx6	0	37.09
invx8	0.84	64.65
invx12	0	118.46
nand2x1	0	4.95
nand2x2	1.47	10.66
nand2x4	2.35	59.54
nand2x6	3.53	116.49
nand3x1	0	17.50
nand3x2	3.07	37.12
nand3x4	7.06	100.98
nand3x6	8.23	197.70
nor2x1	0	3.31
nor2x2	2.65	33.42
nor2x4	2.35	50.44
nor2x6	6.89	93.60
nor3x1	0	17.05
nor3x2	4.70	39.50
nor3x4	6.59	88.90
nor3x6	7.77	118.13

Also we assume that cell delay is a linear function of gate-length for small perturbations of gate-length.

Mixed-Integer Linear Programming (MILP)

Although the sensitivity-based heuristic optimization that uses dense and iso cells for compensating designs results in good solutions, post-processing may be required to ensure the compensation is valid throughout the expected defocus range. Due to the non-linearity of delay (or CD) with focus, an optimization approach should ideally guarantee the resulting solution is valid at all defocus points.

To inherently consider the range of potential defocus conditions, we propose a new optimization approach based on integer linear programming. For each gate i, let the area of component i is Ai, P be the set of all possible paths, and n be the number of gates in circuits. The problem of minimizing total area subject to a maximum delay

bound (required time) can be formulated as [40].

$$\begin{aligned} &\text{Minimize} \quad \textstyle\sum_{i=1}^{n} A_i \\ &\text{Subject to} \quad \textstyle\sum_{i \in p} D_i \le D_{\max} \quad ; \forall p \in P \\ &\qquad A_i \in A_i^{dense}, A_i^{iso} \quad ; i = 1, ..., n \end{aligned} \tag{III.7}$$

The number of possible paths from primary inputs to primary outputs is exponential in n. Therefore, transforming the constraints on path delay into constraints on delay across components (e.g., arrival time) is widely accepted as a practical technique. ai represents the arrival time at each node i while Dmax is the maximum delay bound (required time of the circuit). The above problem can be rewritten as

$$\begin{aligned} &\text{Minimize} \quad \textstyle\sum_{i=1}^{n} A_i \\ &\text{Subject to} \\ &\qquad a_j \le D_{\max} \quad ; j \in outputs \\ &\qquad a_j + D_i \le a_i \quad ; i = 1, ..., n \text{ and } \forall j \in input(i) \\ &\qquad D_i \le a_i \quad ; i = n+1, ...n+s : inputs \\ &\qquad A_i \in A_i^{dense}, A_i^{iso} \quad ; i = 1, ..., n \end{aligned} \tag{III.8}$$

To include the delay variation due to defocus we discretize the defocus into 5 levels (0.0μm, 0.1μm, 0.2μm, 0.3μm, and 0.4um). The ILP problem can then be cast as

$$\begin{aligned} &\text{Minimize} \quad \textstyle\sum_{i=1}^{n} A_i \\ &\text{Subject to} \\ &\qquad a_{j,f} \le D_{\max} \quad ; j \in outputs \\ &\qquad a_{j,f} + D_{i,f} \le a_{i,f} \quad ; i = 1, ..., n \text{ and } \forall j \in input(i) \\ &\qquad D_{i,f} \le a_{i,f} \quad ; i = n+1, ...n+s : inputs \\ &\qquad A_i \in A_i^{dense}, A_i^{iso} \quad ; i = 1, ..., n \\ &\quad f \in \{0.0, 0.1, 0.2, 0.3, 0.4\} \quad ; defocus \end{aligned} \tag{III.9}$$

Where $a_{i,f}$ is the arrival time of gate i at f defocus level, and $D_{i,f}$ represents the gate delay of the ith component at defocus level f.

Finally, given two choices (dense and iso) of gates, the problem can be transformed into an integer (binary) linear optimization problem:

$$
\begin{array}{ll}
\text{Minimize} & \sum_{i=1}^{n} A_i^{dense}(1-x_i) + A_i^{iso}(x_i) \\
\text{Subject to} & \\
& a_{j,f} \leq D_{\max} \qquad ; j \in outputs \\
& a_{j,f} + D_{i,f}^{dense}(1-x_i) + D_{i,f}^{iso}(x_i) \leq a_{i,f} \qquad ; i = 1, ..., n \text{ and } \forall j \in input(i) \\
& D_{i,f}^{dense}(1-x_i) + D_{i,f}^{iso}(x_i) \leq a_{i,f} \qquad ; i = n+1, ...n+s : inputs \\
& x_i \in 0,1 \ \ (binary) \qquad ; i = 1, ..., n(0 = \text{dense}, 1 = \text{iso}) \\
& f \in \{0.0,\ 0.1,\ 0.2,\ 0.3,\ 0.4\} \qquad ; defocus
\end{array}
\tag{III.10}
$$

The integer (binary) linear problem can be efficiently solved using a commercial linear solver. In our case we use the mixed-integer optimizer of CPLEX [164].

Leakage-driven Timing Optimization Leakage is highly sensitive to linewidth variations due to well-known short-channel effects in scaled MOSFETs. Therefore, we propose to perform optimization using dense and iso cells based on leakage characteristics rather than area. A new sensitivity metric that includes the leakage change when an iso cell replaces a dense cell can be formulated as in Equation (III.11) below. We ignore the area change that was considered in Equation (III.6) and instead use *Leak* in the denominator. As can be seen in the Bossung plot (Figure III.8 and Figure III.10), the linewidth of dense cells increases with defocus leading to less leakage. On the other hand, the linewidth of iso cells decreases with defocus, causing dramatically higher leakage in this case. The same heuristic algorithm is applied using this new sensitivity.

$$
Sensitivity = \frac{1}{\Delta Leak} \sum_{arcs} \frac{\Delta D}{slack_{arc} - S_{\min} + K_2} \tag{III.11}
$$

Where *Leak* is the leakage change in switching from dense cells to iso at the worst defocus condition.

III.B.4 Results

To quantify delay variation with defocus across the iso/dense/self-compensated libraries and using our optimization approaches, timing libraries for three different variants of each cell are generated as described earlier. ISCAS benchmark circuits [39] are

then synthesized with the "dense" library at minimum timing constraints using Design Compiler [35].

Table III.8 summarizes the normalized delay and leakage using the various libraries for each benchmark circuit. The table shows that the original library incurs 13% slowdown at worst-case defocus since cells in that library are inherently dense, while the delay decreases 16% on average when using the iso library alone. Both self-compensated and single-pitch cells lead to good robustness across defocus levels. The normalized leakage power information is shown on the right side of the Table III.8. As expected, original and dense libraries at the worst defocus value have much less ($>$ 40%) leakage than the original library at perfect focus since linewidths systematically increase. On the other hand, leakage power with iso cells increase by more than 3X over the original cells at 0.4μm defocus. Leakage power overhead in both self-compensated and single-pitched cells is small (5-6%).

The normalized area overheads incurred when using each cell variant (both uniformly and using the proposed optimization approaches) are shown in Table III.9. The gate distribution and runtime of the optimization options are shown in the right side of the table. Heu1 refers to the heuristic optimization of timing and area and heu2 represents the optimization of timing and leakage described earlier. While self-compensated and single-pitched libraries lead to good timing behavior across focus as already shown, they also lead to relatively large area overheads of 11% and 27% respectively. The ILP optimization provides an optimal solution and can be used to determine how well the heuristics are performing. The two sensitivity-based heuristics show 3-4% area increases while meeting timing requirement throughout all defocus range. Note that the trend is towards smaller area penalties in the larger benchmarks, explainable by the fact that a smaller (relative) subset of gates are responsible for determining timing in these larger circuits. The first heuristic in particular achieves circuit areas very close to optimal, usually within 1%.

In Table III.8 the heuristic optimization considering leakage power results in the use of fewer iso cells than the heuristic based on timing and area since iso cells are being penalized more heavily by leakage than area due to the exponential dependency of the former. However, heuristic 2 still shows a slightly larger area penalty since it is choosing

Table III.7: Top 5 most swapped gates for circuit c5315 by each approach.

heuristic 1 (area)		heuristic 2 (leakage)		ILP (area)	
cells	#of cells	cells	# of cells	cells	#of cells
invx2	298	nand2x6	24	invx12	20
invx8	198	nand2x4	8	invx6	11
nand2x6	19	nor2x4	5	nand2x6	7
nor2x4	7	invx2	4	nand2x4	6
nand2x4	6	nand3x4	3	invx8	2
totals	538		50		47

to exchange gates that show small leakage penalties, which tend to be gates with stacked devices such as NAND2, NAND3, etc. [45]. These gates also are large and incur more severe area penalties when swapped from dense to iso variants. In contrast, heuristic 1 selects very small gates such as inverters to convert to iso since the change in area is being penalized in the sensitivity measure. Table III.7 provides details on the five most commonly swapped gates from dense to iso cells in optimizing the c5315 benchmark using the two different heuristics and the ILP. In line with the above discussion, we observe that there are substantial differences in both the total number of swapping and the type of swapped cells. Despite the fact that heuristic 1 swaps over 10X more cells than heuristic 2 and the ILP solutions, the area penalties are nearly identical for this circuit since most of the swapped cells in heuristic 1 have little to no layout area penalties. As can be seen from the runtime of the various optimization approaches in Table III.8, the heuristic techniques shows very reasonable efficiency with high quality solutions relative to the ILP.

To further illustrate the differences between the heuristic and ILP optimizations, slack vs. defocus is plotted for circuit c7552 in Figure III.15. The graph shows that while the original circuit (based on the original library) fails to meet the required time at defocus, both heuristic and ILP optimization solutions are able to meet the timing requirement throughout the defocus range. In the heuristic optimization, the timing requirement is met both at perfect focus and at the extreme defocus condition initially. However timing failures occur at some intermediate defocus conditions due to the non-linearity of delay and focus. The post-processing step described in can handle

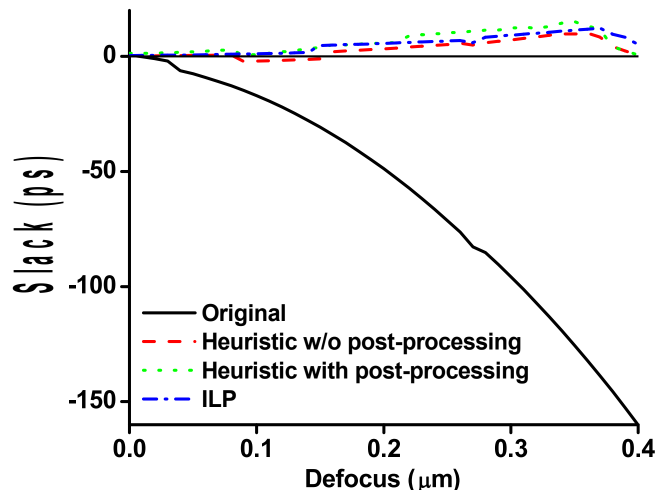


Figure III.15: Slack vs. defocus for benchmark c7552 showing the effectiveness of various self-compensating design options. Note some defocus values (e.g., 0.1-0.18μm) at which the circuit fails to meet timing requirement under the heuristic optimization without post-processing. The horizontal line at y=0 is added to highlight the timing constraint.

the problems and guarantee the positive slack in all defocus range. From the results of Table III.8 and Figure III.15, we observe that the sensitivity-based heuristics with post-processing are very close to the ILP results. Therefore, we do not show the results of running the ILP formulation with the leakage objective instead of area.

A Monte-Carlo simulation with 1000 trials is employed to investigate the impact of defocus variation on delay distribution. A normal distribution of focus with mean = 0.0μm and 3σ = 0.4μm is assumed. Figure III.16 shows Monte-Carlo simulation results for the c6288 benchmark. Self-compensated, single-pitch, and the two dense + iso optimization options meet the timing requirement at all 1000 randomly chosen defocus points.

Table III.10 shows the change in leakage power at the worst defocus conditions compared to the original library at perfect focus using several self-compensating design options. As can be seen, both self-compensated cells and single-pitch cells designs options shows modest ~7% leakage increases at worst-case defocus. The area-driven dense + iso optimization shows 10% less leakage than the nominal case at 0.4μm defocus, although

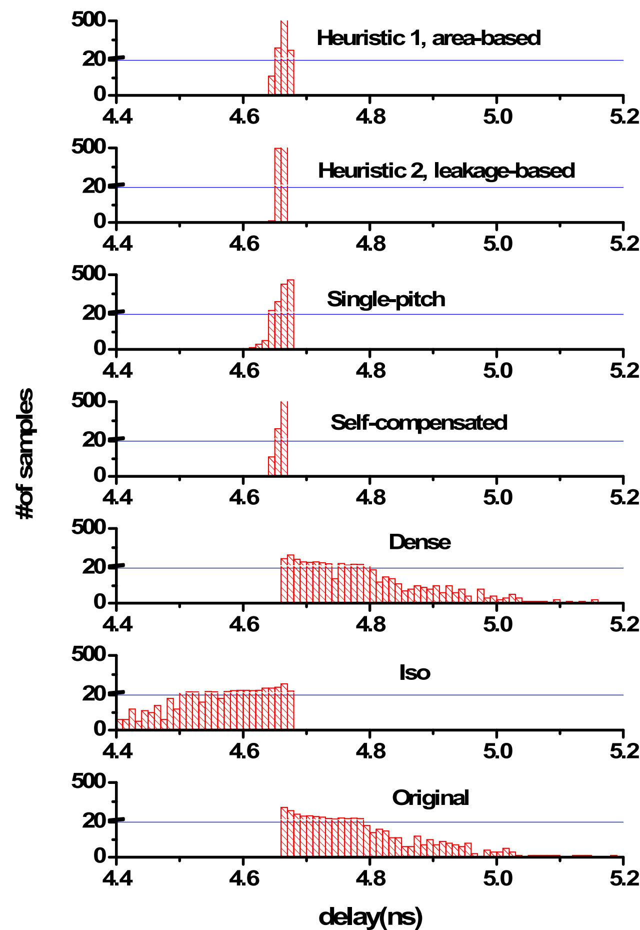


Figure III.16: Stacked histograms showing the delay distribution for c6288 (required time = 4.68ns). Note that there is a break in the y-axis at 21.

the results for this case vary widely. As expected the leakage-driven optimization shows 25% less leakage than the original circuit and 15% less than heuristic 1 since leakage is directly accounted for in this formulation.

III.B.5 Conclusions

A novel design technique to compensate for lithographic focus-dependent CD variation is proposed in this chapter. The general idea is to judiciously instantiate isolated and dense versions of library cells in a circuit to effectively negate the impact of expected focus variations. We present two heuristic approaches to self-compensated design

for focus-dependent CD variation along with an ILP formulation. All three algorithms lead to circuits that can meet timing requirements across expected defocus levels while incurring only very small area penalties. Specifically we can achieve a focus compensated design with ~3% area overhead, compared to 11% and 27% in a self-compensated and single-pitch library based design, respectively. In addition, we investigate the leakage impact of defocus and one of the heuristics seeks to minimize leakage while meeting timing requirements. Results using both iso and dense libraries together show 30% lower leakage compared to circuits designed using an inherently self-compensated library under worst-case focus conditions.

III.C Acknowledgments

Chapter III is in part of reprint of "Self-Compensating Design for Reduction of Timing and Leakage Sensitivity to Systematic pattern-dependent Variation", to appear in *IEEE Transactions on CAD* and "Toward a Systematic-Variation Aware Timing Methodology", *Proc. ACM/IEEE Design Automation Conference*, 2004. I would like to thank by coauthors Youngmin Kim, Dr. Dennis Sylvester, Dr. Fook-Luen Heng and Dr. Andrew B. Kahng.

Table III.8: Normalized delay and leakage power for ISCAS85 benchmark circuits synthesized in each library type (normalized to original cells at 0.0μm defocus value).

benchmarks	original 0μm defocus	**normalized delay**					**normalized leakage**				
		0.4μm defocus					0.4μm defocus				
		original	iso	dense	self-comp.	single-pitch	original	iso	dense	self-comp.	single-pitch
c432	1.00	1.13	0.84	1.13	0.99	0.99	0.57	3.17	0.57	1.05	1.07
c499	1.00	1.12	0.83	1.13	1.00	0.99	0.58	3.17	0.57	1.06	1.07
c880	1.00	1.12	0.84	1.12	1.00	0.99	0.57	3.20	0.57	1.06	1.07
c1355	1.00	1.12	0.84	1.12	1.00	0.99	0.58	3.15	0.57	1.06	1.07
c1908	1.00	1.13	0.84	1.13	1.00	0.99	0.58	3.12	0.57	1.05	1.06
c2670	1.00	1.14	0.83	1.13	0.99	0.99	0.58	3.12	0.58	1.05	1.06
c3540	1.00	1.12	0.84	1.13	0.99	0.99	0.57	3.18	0.57	1.05	1.07
c5315	1.00	1.13	0.83	1.14	0.99	0.99	0.58	3.11	0.57	1.05	1.06
c6288	1.00	1.13	0.83	1.13	1.00	0.99	0.58	3.14	0.57	1.05	1.06
c7552	1.00	1.12	0.83	1.13	1.00	0.99	0.58	3.09	0.57	1.05	1.06
Average	**1.00**	**1.13**	**0.84**	**1.13**	**0.99**	**0.99**	**0.58**	**3.15**	**0.57**	**1.05**	**1.06**

Table III.9: Normalized area and gate distribution for each library and optimization approach.

Benchmark	Total #gates	normalized area								gate distribution						Runtime (sec)	
		orig.	dense	iso	self comp.	single pitch	heu1 (area)	heu2 (leakage)	ILP	heu1 (area)		heu2(leakage)		ILP			
										dense	iso	dense	iso	dense	iso	heu1	ILP
c432	339	1.00	1.02	1.17	1.12	1.26	1.09	1.09	1.08	233	106	318	21	317	22	0.04	0.19
c499	682	1.00	1.00	1.17	1.11	1.27	1.00	1.02	1.00	581	101	569	113	584	98	0.09	1.70
c880	575	1.00	1.02	1.18	1.11	1.27	1.02	1.02	1.01	560	15	561	14	562	13	0.07	0.35
c1355	680	1.00	1.00	1.17	1.11	1.27	1.05	1.08	1.04	536	144	516	164	564	116	0.39	11.21
c1908	645	1.00	1.01	1.16	1.12	1.26	1.04	1.05	1.04	554	91	584	61	566	79	0.08	13.79
c2670	1040	1.00	1.01	1.15	1.11	1.25	1.05	1.05	1.04	1017	23	1020	20	1010	30	0.20	11.61
c3540	1313	1.00	1.01	1.17	1.10	1.27	1.01	1.01	1.01	1279	34	1287	26	1280	33	0.32	27.28
c5315	2028	1.00	1.00	1.16	1.11	1.27	1.01	1.01	1.00	1490	538	1978	50	1981	47	1.51	29.30
c6288	4102	1.00	1.00	1.16	1.11	1.26	1.06	1.07	1.05	3631	471	3820	282	3693	409	7.80	913.32
c7552	2700	1.00	1.00	1.15	1.11	1.25	1.01	1.01	1.00	2610	90	2658	42	2648	52	2.02	358.03
average		**1.00**	**1.01**	**1.16**	**1.11**	**1.27**	**1.03**	**1.04**	**1.02**								

Table III.10: Leakage power change for self-compensating designs and two heuristic-based optimizations at 0.4μm defocus compared to the original library at 0.0μm defocus.

at 0.4 defocus	c432	c499	c880	c1355	c1908	c2670	c3540	c5315	c6288	c7552	Avg.
self-compensated	5.1%	5.7%	5.5%	5.6%	5.5%	5.0%	5.5%	5.5%	5.3%	5.4%	**5.4%**
single-pitched	6.6%	6.9%	7.0%	6.7%	6.2%	5.9%	6.8%	6.2%	6.4%	5.9%	**6.5%**
heu1 (area)	31.4%	-4.8%	-36.3%	11.3%	-6.5%	-36.5%	-34.5%	17.4%	-10.5%	-33.6%	**-10.3%**
heu2 (leakage)	-25.7%	0.6%	-36.9%	14.8%	-22.2%	-37.6%	-37.2%	-36.6%	-25.9%	-39.0%	**-24.6%**

IV

Dealing With FEOL Leakage Variability by Gate-Length Biasing

High power dissipation in integrated circuits shortens battery life, reduces circuit performance and reliability, and has a large impact on packaging cost. Power in CMOS circuits consists of dynamic and static (due to leakage currents) components. Leakage is becoming an ever-increasing component of total dissipated power, with its contribution projected to increase from 18% at 130nm to 54% at the 65nm node [73]. Leakage is composed of three major components: (1) subthreshold leakage, (2) gate leakage, and (3) reverse biased drain substrate and source-substrate junction band-to-band tunneling leakage [58]. Subthreshold leakage is the dominant contributor to total leakage at 130nm and is forecast to remain so in the future [58]. In this work we present a novel approach for subthreshold leakage reduction.

Leakage reduction methodologies can be divided into two classes depending on whether they reduce *standby* leakage or *runtime* leakage. Standby techniques reduce leakage of devices that are known not to be in operation while runtime techniques reduce leakage of active devices. Several techniques have been proposed for standby leakage reduction. *Body biasing* or *VTMOS* based approaches [64] dynamically adjust the device

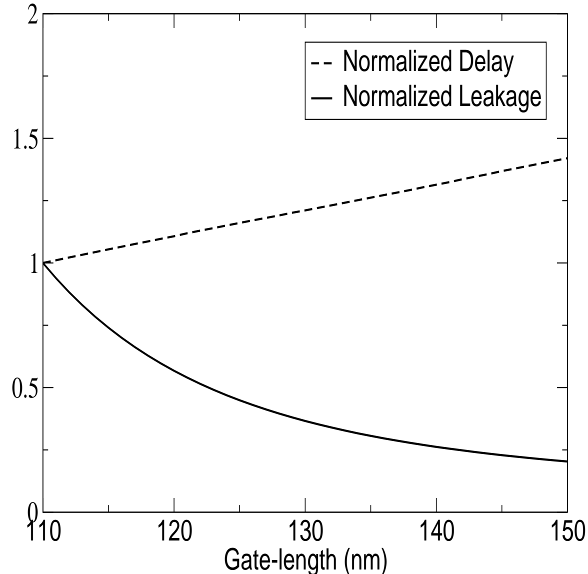


Figure IV.1: Variation of leakage and delay (each normalized to 1.00) for an NMOS device in an industrial 130nm technology.

V_{th} by biasing the body terminal [1]. *Muti-threshold CMOS (MTCMOS)* techniques [69, 65, 70, 78] use high-V_{th} CMOS (or NMOS or PMOS) to disconnect Vdd or Vss or both to logic circuit implemented using low V_{th} devices in standby mode. *Source biasing*, where a positive bias is applied in standby state to source terminals of off devices, was proposed in [63]. Other techniques such as use of transistor stacks [45] and input-vector control [62] have also been proposed.

The only mainstream approach to runtime leakage reduction is the multi-V_{th} manufacturing process. In this approach, cells in non-critical paths are assigned a high V_{th} while cells in critical paths are assigned a low V_{th}. [82] presented a heuristic algorithm for selection and assignment of an optimal high V_{th} to cells on non-critical paths. The multi-V_{th} approach has also been combined with several other power reduction techniques [67, 84, 79]. The primary drawback to this technique has traditionally been the rise in process costs due to additional steps and masks. However, the increased costs have been outweighed by the resulting substantial leakage reductions, and multi-V_{th} processes are now standard. A new complication facing multi-V_{th} is the increased variability of V_{th} for low-V_{th} devices. This occurs in part due to random doping fluctuations, as well as wors-

[1] Body biasing has also been proposed to reduce leakage of active devices [74].

ened DIBL (Drain Induced Barrier Lowering) and short-channel effects (SCE) in devices with lower channel doping. The larger variability in V_{th} degrades the achievable leakage reductions of multi-V_{th} and worsens with continued MOS scaling. Moreover, multi-V_{th} methodologies do not offer a smooth tradeoff between performance and leakage power. Devices with different V_{th} typically have a large separation in terms of performance and leakage, for instance a 15% speed penalty with a 10× reduction in leakage for high-V_{th} devices.

The use of longer gate-lengths ($Lgate$) in devices within non-critical gates was first described in [81]. In that work, *large* changes to gate-lengths were considered, resulting in heavy delay and dynamic power penalties. Moreover, cell layouts with significantly larger gate-lengths are not layout-swappable with their nominal versions, resulting in substantial ECO (Engineering Change Order) overheads during layout. In this chapter, we propose very *small* increases in gate-length for non-critical devices. These small increases maximize the leakage reduction since they take full advantage of the SCE and incur only very small penalties in drive current and input capacitance. Technologies at the 90nm node and below employ super-halo doping, giving rise to reverse short channel effects (RSCE) that mitigate traditional SCE to some extent. However, we have found the proposed technique to substantially reduce leakage for the two 130nm and two 90nm industrial processes that we investigated. Recent reports from leading integrated device manufacturers (IDMs) indicate SCE continues to dominate V_{th} roll-off characteristics at the 65nm and 45nm technology nodes [71, 72, 60, 68]. However, we note that the V_{th} roll-off curve must be understood to assess the feasibility of this approach and to determine reasonable increases for gate length.

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

Contributions of our work include the following.

- A leakage reduction methodology based on less than 10% increase in drawn *Lgate* of devices.
- A thorough analysis of potential benefits and caveats of such a biasing methodology, including implications of lithography and process variability.
- Experiments and results showing potential benefits of an *Lgate* biasing methodology in different design scenarios such as dual-V_{th}.

IV.A Cell-Level Gate-Length Biasing

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

IV.A.1 Library Generation

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

- The nominal gate-length of the technology is usually very close to or beyond the "knee" of the leakage vs. *Lgate* curve which arises due to SCE. For large bias, the advantage of super-linear dependence of leakage on gate-length is lost. Moreover, dynamic power and delay both increase almost linearly with gate-length. Therefore, small biases give more "bang for the buck".
- From a manufacturability point of view (discussed later in Section IV.B.2), having two prevalent pitches (which are relatively distinct) in the design can harm printability properties (i.e., size of process window). We retain the same poly-pitch

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

as the unbiased version of the cell: there is a small decrease in spacing between gate-poly geometries.

- An increase in drawn dimension that is less than the layout grid resolution (typically $10nm$ for $130nm$ technology) ensures pin-compatibility with the unsized version of the cell. This is very important to ensure that multi-$Lgate$ optimizations can be done post-placement or even after detailed-routing without ECOs. In this way, we retain the layout transparency that has made multi-V_{th} optimization so adoptable within chip implementation flows. Biases smaller than the layout grid-pitch also ensure design-rule correctness for the biased cell layout, provided that the unbiased version is design-rule correct.

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

An important component of the methodology is layout and characterization of the dual-$Lgate$ library. Since we investigate very small biases to the gate-length, the layout of the biased library cell does not need to change except for a simple automatic scaling of dimensions. Of course, after the slight modifications to the layout, the biased versions of the cell are put through the standard extraction and power/timing characterization process.

IV.A.2 Optimization for Leakage

We perform standard gate sizing (gate-width sizing) prior to *Lgate* biasing using *Synopsys Design Compiler v2003.06-SP1.* Since delay is almost always the primary design goal, we perform sizing to achieve the minimum possible delay. We use a sensitivity-based, *downsizing* (i.e., begin with all nominal cell variants and replace cells on non-critical paths with biased variants) algorithm for leakage optimization. In our studies, we have found downsizing to be significantly more effective at leakage reduction than *upsizing* (i.e., begin with all biased variants in the circuit and replace critical cells with their nominal-*Lgate* variants) irrespective of the delay constraints. An intuitive rationale is that upsizing approaches have dual objectives of delay and leakage during cell selection for upsizing. Downsizing approaches, on the other hand, only downsize cells that do not cause timing violations and have the sole objective of leakage minimization. We note that an upsizing approach, however, may be faster when loose delay constraints are to be met since very few transistors have to be upsized. However, delay is almost always the primary design goal and loose delay constraints are rare. A timing analyzer is an essential component of any delay-aware power optimization approach; it is used to compute delay sensitivity to biasing of cell instances in the design. For an accurate yet scalable implementation, we use three types of timers that vary in speed and accuracy.

- *Standard static timing analysis (SSTA).* Slews and actual arrival times (AATs) are propagated forward after a topological ordering of the circuit. Required arrival times (RATs) are back-propagated and slacks are then computed. Slew, delay and slack values of our timer match exactly with *Synopsys PrimeTime vU-2003.03-SP2* and our timer can handle unate and non-unate cells [3].

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

[3] Delay values from our timer match with PrimeTime only under our restricted use model. Our timer does not support several important features such as interconnect delay, hold time checks, false paths, multiple clocks, 3-pin SDFs, etc.

- *Constrained incremental STA (CISTA).* Sensitivity computation involves temporary modifications to a cell to find changes in its slack and leakage. To make this step faster, we restrict the incremental timing calculation to only one stage before and after the gate being modified. The next stage is affected by slew changes and the previous stage is affected by the pin capacitance change of the modified gate. The ripple effect on other stages farther away from the gate (primarily due to slew changes[4]) is neglected since high accuracy is not critical for sensitivity computation.

We use the phrase "downsizing a cell instance" (or node) to mean replacing it by its biased variant in the circuit. In our terminology, s_p represents the slack on a given cell instance p, and s'_p represents the slack on p after it has been downsized. ℓ_p and ℓ'_p indicate the initial and final leakages of cell instance p before and after downsizing respectively. P_p represents the sensitivity associated with cell instance p and is defined as:

$$P_p = \frac{\ell_p - \ell'_p}{s_p - s'_p}$$

The pseudocode for our leakage optimization implementation is given in Figure IV.2. The algorithm begins with SSTA and initializes slack values s_p in Line 1. Sensitivities P_p are computed for all cell instances p and put into a set S in Lines 2-5. We select and remove the largest sensitivity P_{p^*} from the set S and continue with the algorithm if $P_{p^*} \geq 0$. In Line 11, the function *SaveState* saves the gate-lengths of all transistors in the circuit as well as the delay, slew and slack values. The cell instance p^* is downsized and EISTA is run from it to update the delay, slew and slack values in Lines 12-13. Our timing libraries capture the effect of biasing on slew as well as input capacitance, and our static timing analyzer efficiently and accurately updates the design to reflect the changes in delay, capacitance and slew due to the downsizing move. If there is no timing violation (negative slack on any timing arc) then this move is accepted, otherwise the saved state is restored. If the move is accepted, we also update

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

```
procedure LGateBiasing
1 Run STA to initialize s_p  ∀cell instances, p
2 S ← {}
3 forall cell instances, p
4  P_p ← ComputeSensitivity(p)
5  S ← S ∪ P_p
6 do
7  P_p* ← max(S)
8  if(P_p* ≤ 0)
9   exit
10  S ← S − {P_p*}
11  SaveState()
12  Downsize cell instance p*
13  EISTA(p*)
14  if(TimingViolated())
15   RestoreState()
16  else
17   N ← p* ∪ fan-in and fan-out nodes of p*
18   forall q ∈ N
19    if(P_q ∈ S)
20     P_q ← ComputeSensitivity(q)
21     Update P_q in S
22 while(|S| > 0)
procedure ComputeSensitivity(q)
1 old_slack ← Slack on cell instance q
2 old_leakage ← Leakage of cell instance q
3 SaveState()
4 Downsize cell instance q
5 CISTA(q)
6 new_slack ← Slack on cell instance q
7 new_leakage ← Leakage of cell instance q
8 RestoreState()
9 return (old_leakage − new_leakage)/(old_slack − new_slack)
```

Figure IV.2: Pseudocode for cell-level gate-length biasing for leakage optimization.

sensitivities of node p^*, its fan-in fan-out nodes in Lines 17-21. The algorithm continues until the largest sensitivity becomes negative or the size of S becomes zero. Function $ComputeSensitivity(q)$ temporarily downsizes cell instance q and finds its slack using CISTA. Since high accuracy is not critical for sensitivity computation we choose to use CISTA which is faster but less accurate than EISTA. Table IV.1 shows a comparison of leakage and runtime when EISTA and CISTA are used for sensitivity computation.

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

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

Table IV.2: Test cases used in our experiments and their details.

Test Case	Source	#Cells	Delay (ns)	Leakage (mW)	Dynamic (mW)
s9234	ISCAS'89	861	0.437	0.7074	0.3907
c5315	ISCAS'85	1442	0.556	1.4413	1.5345
c7552	ISCAS'85	1902	0.485	1.8328	2.0813
s13207	ISCAS'89	1957	0.904	1.3934	0.6296
c6288	ISCAS'85	4289	2.118	3.5994	8.0316
alu128	Opencores.org [163]	7536	2.306	5.1571	4.4177
s38417	ISCAS'89	7826	0.692	4.9381	4.2069

IV.B Experiments and Results

We now describe our test flow for validation of the *Lgate* biasing methodology, and present experimental results. Details of the test cases[5] used in our experiments are given in Table IV.2. The test cases are synthesized with the *Artisan TSMC* 130*nm* *library* using *Synopsys Design Compiler v2003.06-SP1* with low-V_{th} cells only. To limit library characterization runtime, we restrict the library to variants of the following 25 most frequently used cells: CLKINVX1, INVX12, INVX1, INVX3, INVX4, INVX8, INVXL, MXI2X1, MXI2X4, NAND2BX4, NAND2X1, NAND2X2, NAND2X4, NAND2X6, NAND2X8, NAND2XL, NOR2X1, NOR2X2, NOR2X4, NOR2X6, NOR2X8, OAI21X4, XNOR2X1, XNOR2X4, XOR2X4. To identify the most frequently used cells, we synthesize our test cases with the complete library and select the 25 most frequently used cells.

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

The delay constraint is kept tight so that the post-synthesis delay is close to minimum achievable delay.

We consider up to two gate-lengths and two threshold voltages. We perform experiments for the following scenarios: (1) Single-V_{th}, single-*Lgate* (SVT-SGL), (2) Dual-V_{th}, single *Lgate* (DVT-SGL), (3) Single-V_{th}, dual-*Lgate* (SVT-DGL), and (4) Dual-V_{th}, dual *Lgate* (DVT-DGL). The dual-V_{th} flow uses nominal and low values of V_{th} while the single-V_{th} flow uses only the low value of V_{th}. *STMicroelectronics* 130*nm* device models are used with the two V_{th} values each for PMOS and NMOS transistors (PMOS: -0.09V and -0.17V; NMOS: 0.11V and 0.19V). We use *Cadence SignalStorm v4.1* (with *Synopsys HSPICE*) for delay and power characterization of cell variants. *Synopsys Design Compiler* is used to measure circuit delay, dynamic power and leakage power. We assume an activity factor of 0.02 for dynamic power calculation in all our experiments. We do not assume any wire-load models, as a result of which the dynamic power and delay overheads of *Lgate* biasing are conservative (i.e., overestimated). All experiments are run on an Intel Xeon 1.4GHz computer with 2GB of RAM.

IV.B.1 Leakage Reduction

Table IV.3 shows the leakage savings and delay penalties due to *Lgate* biasing for all cells in our library. The results strongly support our hypothesis that small biases in *Lgate* can afford significant leakage savings with small performance impact. To assess the maximum impact of biasing, we explore the power-performance envelope obtained by replacing every device in the design by its device-level biased variant.

We now use our leakage optimization approach to selectively bias cells on non-critical paths. Table IV.4 shows the leakage reduction, dynamic power penalty, and total power reduction for our test cases when *Lgate* biasing is applied without dual-V_{th} assignment. Table IV.5 shows results when *Lgate* biasing is applied together with the dual V_{th} approach. To show the effectiveness of *Lgate* biasing with loose delay constraints, results when the delay constraint is relaxed are also shown for each circuit. The leakage reductions primarily depend on the slack profile of the circuit. If a large number of paths have near-zero slacks then the leakage reductions are smaller. As the delay penalty increases more slack is introduced on paths and larger leakage reductions

Table IV.3: Leakage reduction and delay penalty due to gate-length biasing for all 25 cells in our library.

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

are seen. We observe that leakage reductions are smaller when the circuit has already been optimized using dual-V_{th} assignment. This is expected because dual-V_{th} assignment consumes slack on non-critical paths reducing the slack available for $Lgate$ optimization. We also observe larger leakage reductions in sequential circuits; this is because circuit delay is determined by the slowest pipeline stage and the percentage of non-critical paths is typically higher in sequential circuits.

Our leakage models do not include gate leakage, which can marginally increase due to biasing. Gate leakage is composed of gate-length-dependent (gate-to-channel (I_{gc})

Table IV.4: Impact of gate-length biasing on leakage and dynamic power (assuming an activity factor of 0.02) for single threshold-voltage designs. Delay penalty constraint is set to 0%, 2.5%, and 5% for each of the test cases. (Note: delay penalty for SVT-SGL is always set to 0% due to the non-availability of V_{th} and $Lgate$ knobs. SVT-DGL is slower than SVT-SGL for delay penalties of 2.5% and 5%.)

Test	Delay (ns)	SVT-SGL			SVT-DGL			Reduction			CPU (s)
		Leakage (mW)	Dynamic (mW)	Total (mW)	Leakage (mW)	Dynamic (mW)	Total (mW)	Leakage (%)	Dynamic (%)	Total (%)	
s9234	0.437	0.7074	0.3907	1.0981	0.5023	0.4005	0.9028	28.99	-2.50	17.79	1.81
	0.447	0.7074	0.3907	1.0981	0.5003	0.4006	0.9008	29.28	-2.52	17.96	1.79
	0.458	0.7074	0.3907	1.0981	0.4983	0.4006	0.8988	29.56	-2.51	18.15	1.79
c5315	0.556	1.4413	1.5345	2.9758	1.2552	1.5455	2.8007	12.91	-0.72	5.88	5.60
	0.570	1.4413	1.5345	2.9758	1.0415	1.5585	2.6000	27.74	-1.56	12.63	5.80
	0.584	1.4413	1.5345	2.9758	1.0242	1.5604	2.5846	28.94	-1.69	13.15	5.79
c7552	0.485	1.8328	2.0813	3.9141	1.4447	2.0992	3.5439	21.18	-0.86	9.46	10.97
	0.497	1.8328	2.0813	3.9141	1.3665	2.1042	3.4707	25.44	-1.10	11.33	11.08
	0.509	1.8328	2.0813	3.9141	1.3177	2.1084	3.4261	28.10	-1.30	12.47	10.89
s13207	0.904	1.3934	0.6296	2.0230	0.9845	0.6448	1.6293	29.35	-2.42	19.46	11.46
	0.927	1.3934	0.6296	2.0230	0.9778	0.6449	1.6226	29.83	-2.42	19.79	11.31
	0.949	1.3934	0.6296	2.0230	0.9758	0.6446	1.6204	29.97	-2.39	19.90	11.27
c6288	2.118	3.5994	8.0316	11.6310	3.3391	8.0454	11.3845	7.23	-0.17	2.12	70.51
	2.171	3.5994	8.0316	11.6310	2.8461	8.0931	10.9392	20.93	-0.77	5.95	74.79
	2.224	3.5994	8.0316	11.6310	2.7415	8.1051	10.8466	23.83	-0.92	6.74	70.11
alu128	2.306	5.1571	4.4177	9.5748	4.5051	4.4429	8.9480	12.64	-0.57	6.55	270.00
	2.363	5.1571	4.4177	9.5748	3.5992	4.4818	8.0810	30.21	-1.45	15.60	212.97
	2.421	5.1571	4.4177	9.5748	3.5900	4.4826	8.0726	30.39	-1.47	15.69	211.47
s38417	0.692	4.9381	4.2069	9.1450	3.4847	4.2765	7.7612	29.43	-1.65	15.13	225.18
	0.710	4.9381	4.2069	9.1450	3.4744	4.2778	7.7522	29.64	-1.69	15.23	225.68
	0.727	4.9381	4.2069	9.1450	3.4713	4.2779	7.7492	29.70	-1.69	15.26	221.35

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

Test	Delay (ns)	DVT-SGL			DVT-DGL			Reduction			CPU (s)
		Leakage (mW)	Dynamic (mW)	Total (mW)	Leakage (mW)	Dynamic (mW)	Total (mW)	Leakage (%)	Dynamic (%)	Total (%)	
s9234	0.437	0.0984	0.3697	0.4681	0.0722	0.3801	0.4523	26.60	-2.81	3.37	1.86
	0.447	0.0914	0.3691	0.4604	0.0650	0.3798	0.4448	28.81	-2.90	3.39	1.89
	0.458	0.0873	0.3676	0.4549	0.0609	0.3784	0.4393	30.20	-2.95	3.41	1.83
c5315	0.556	0.3772	1.4298	1.8070	0.3391	1.4483	1.7874	10.11	-1.29	1.09	5.74
	0.570	0.2871	1.4193	1.7064	0.2485	1.4390	1.6875	13.45	-1.39	1.11	6.21
	0.584	0.2401	1.4119	1.6520	0.1986	1.4328	1.6314	17.27	-1.48	1.24	6.14
c7552	0.485	0.6798	1.9332	2.6130	0.6655	1.9393	2.6048	2.10	-0.32	0.31	10.40
	0.497	0.4698	1.9114	2.3812	0.4478	1.9210	2.3689	4.68	-0.50	0.52	10.51
	0.509	0.3447	1.8994	2.2441	0.3184	1.9107	2.2291	7.63	-0.59	0.67	10.55
s13207	0.904	0.1735	0.5930	0.7664	0.1247	0.6069	0.7316	28.09	-2.35	4.54	11.59
	0.927	0.1561	0.5920	0.7481	0.1066	0.6060	0.7127	31.68	-2.37	4.73	11.73
	0.949	0.1536	0.5919	0.7455	0.1027	0.6060	0.7087	33.14	-2.39	4.93	11.76
c6288	2.118	1.9733	7.7472	9.7205	1.9517	7.7572	9.7089	1.09	-0.13	0.12	79.25
	2.171	1.2258	7.5399	8.7657	1.1880	7.5574	8.7454	3.08	-0.23	0.23	79.25
	2.224	0.8446	7.4160	8.2606	0.8204	7.4283	8.2487	2.87	-0.17	0.14	77.28
alu128	2.306	0.6457	3.9890	4.6347	0.5184	4.0353	4.5537	19.73	-1.16	1.75	240.09
	2.363	0.6151	3.9837	4.5988	0.4970	4.0242	4.5212	19.21	-1.02	1.69	262.37
	2.421	0.5965	3.9817	4.5782	0.4497	4.0378	4.4875	24.62	-1.41	1.98	277.99
s38417	0.692	0.5862	3.8324	4.4186	0.4838	3.8680	4.3518	17.46	-0.93	1.51	238.62
	0.710	0.5637	3.8309	4.3946	0.4189	3.8861	4.3050	25.69	-1.44	2.04	238.99
	0.727	0.5504	3.8306	4.3810	0.4067	3.8849	4.2916	26.11	-1.42	2.04	234.94

and gate-to-body (I_{gb}) tunneling) and independent components (edge direct tunneling ($I_{gs} + I_{gd}$)). The gate-length-independent component, which stems from the gate-drain and gate-source overlap regions, is not affected by biasing. To assess the change in gate-length-dependent components due to biasing we perform SPICE simulations to report the gate-to-channel leakage[6] for nominal and biased devices. We use 90nm *BSIM4* device models from a leading foundry that model all five components of gate leakage described in *BSIM v4.4.0*. Table IV.6 shows the gate and subthreshold leakage for biased and unbiased nominal V_{th} NMOS and PMOS devices of 1μm width at 25^oC and 125^oC. The reductions in subthreshold and gate leakage as well as the total leakage reduction are shown. Based on these results, we conclude that the increase in gate leakage due to biasing is negligible. Furthermore, since biasing is a runtime leakage reduction approach, the operating temperature is likely to be higher than room temperature – in this scenario gate leakage is not a major portion of total leakage. When the operating temperature is elevated, the reduction in total leakage is approximately equal to the reduction in subthreshold leakage and total leakage reductions similar to the results presented in Tables IV.4 and IV.5 are expected[7]. Gate leakage is predicted to increase with technology scaling; technologies under 65nm, however, are likely to adopt high-k gate dielectrics which will tremendously reduce gate leakage so in terms of scalability, subthreshold leakage remains the key problem at high operating temperatures. We also note that because the vertical electric fields do not increase due to biasing, negative-bias thermal instability (NBTI) is not expected to increase with biasing [77].

IV.B.2 Manufacturability and Process Effects

In this subsection, we investigate the manufacturability and process variability implications of our *Lgate* biasing approach. As our method relies on biasing of drawn gate-length, it is important to correlate this with actual printed gate-length on the wafer. This is even more important as the bias we introduce in gate-length is of the same order as the typical critical dimension (CD) tolerances in manufacturing processes. Moreover,

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

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

Table IV.6: Impact of gate-length biasing on subthreshold leakage and gate tunneling leakage of $90nm$ PMOS and NMOS devices of $1\mu m$ width at different temperatures. Total leakage reductions are high even when gate leakage is considered.

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

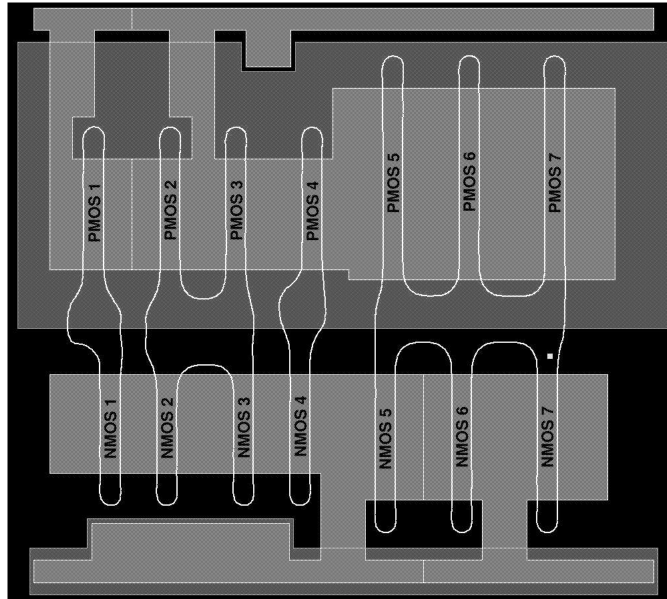


Figure IV.3: Cell layout of a generic AND2X6 with simulated printed gate-lengths.

we expect larger gate-lengths to have better printability properties leading to less CD - and hence leakage - variability. To validate our multiple gate-length approach in a post-manufacturing setup, we follow a reticle enhancement technology (RET) and process simulation flow for an example cell master.

We use the layout of a generic AND2X6 cell and perform model-based optical proximity correction (OPC) on it using *Calibre v9.3_2.5* [162].[8] The printed image of the cell is then calculated using *dense* simulation in Calibre. The layout of the cell along with printed gate-lengths of all devices in it is shown in Figure IV.3. We measure the *Lgate* for every device in the cell, for both biased and unbiased versions. The printed gate-lengths for the seven NMOS and PMOS devices labeled in Figure IV.3 are shown in Table IV.7. As expected, biased and unbiased gate-lengths track each other well. There are some outliers that may be due to the relative simplicity of the OPC model being used. High correlation between *printed* dimensions of biased and unbiased versions of the cells shows that the benefits of biasing estimated using *drawn* dimensions will not be lost after RET application and the manufacturing process.

Another potentially valuable benefit of slightly larger gate-lengths is the pos-

[8]Model-based OPC is performed using annular optical illumination with $\lambda = 248nm$ and $NA = 0.7$.

Table IV.7: Comparison of printed dimensions of unbiased and biased versions of AND2X6. The unbiased nominal gate-length is 130nm while the biased nominal is 138nm. Note the high correlation between unbiased and biased versions.

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

Table IV.8: Process window improvement with gate-length biasing. The CD tolerance is kept at 13nm. ELAT=Exposure latitude.

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

sibility of improved printability. Minimum poly spacing is larger than poly gate-length, so that the process window (which is constrained by the minimum resolvable dimension) tends to be larger as gate-length increases even though poly spacing decreases. For example, the depth of focus for various values of exposure latitude with the same illumination system as above for 130nm and 138nm lines is shown in Table IV.8.[9]

IV.B.3 Process Variability

A number of sources of variation can cause fluctuations in gate-length, and hence in performance and leakage. This has been a subject of much discussion in the recent literature (e.g., [75, 26]). Up to 20× variation in leakage has been reported in production microprocessors [116]. For leakage, the reduction in variation post-biasing

[9]The process simulation was performed using *Prolith v8.1.2* [57].

Table IV.9: Reduction in performance and leakage power uncertainty with biased gate-length in the presence of inter-die variations. The uncertainty spread is specified as a percentage of nominal. The results are given for dual V_{th} and the biasing is $8nm$.

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

is likely to be substantial as the larger gate-length is closer to the "flatter" region of the V_{th} vs. $Lgate$ curve. To validate this intuition, we study the impact of gate-length variation on leakage and performance both pre- and post-biasing using a simple worst-case approach. We assume the CD variation budget to be $\pm 10nm$. The performance and leakage of the test case circuits is measured at the worst-case, nominal and best-case process corners which consider just gate-length variation. This is done for the DVT-DGL approach in which biasing is done along with dual V_{th} assignment. The results are shown in Table IV.9. For the seven test cases, we see up to a 41% reduction in leakage power uncertainty caused by linewidth variation. Such large reductions in uncertainty can potentially outweigh benefits of alternative leakage control techniques. We note that the corner case analysis only models the inter-die component of variation, which typically constitutes roughly half of the total CD variation.

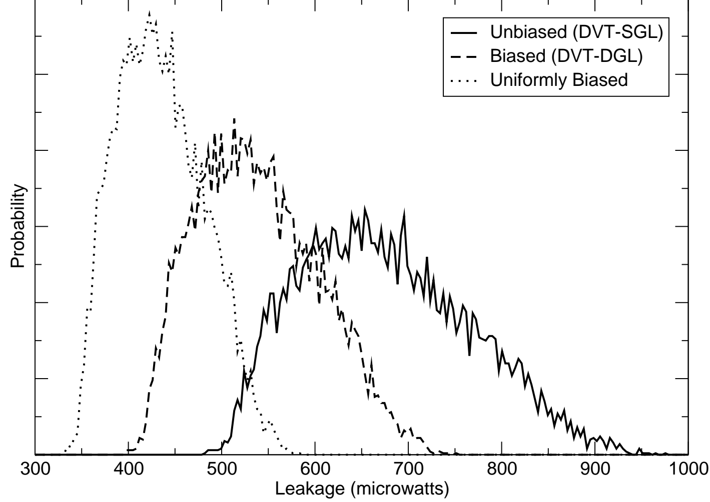


Figure IV.4: Leakage distributions for unbiased, uniform-biased and technology-level selectively-biased alu128. Note the "left-shift" of the distribution with the introduction of biased devices in the design.

To assess the impact of both within-die (WID) and die-to-die (DTD) components of variation, we run 10,000 Monte-Carlo simulations with $\sigma_{WID} = \sigma_{DTD} = 3.33nm$. The variations are assumed to follow a Gaussian distribution with no correlations. We compare the results for three dual V_{th} scenarios: unbiased (DVT-SGL), biased (DVT-DGL) and uniformly biased (when gate-lengths of all transistors in the design are biased by $8nm$). Leakage distributions for the test case $alu128$ are shown in Figure IV.4. Note that in uniform biasing, all devices are biased and the circuit delay no longer meets timing.

IV.C Conclusions and Ongoing Work

We have presented a novel methodology that uses selective, *small Lgate* biases to achieve an *easily manufacturable* approach to runtime leakage reduction. For our test cases we have observed the following.

- The gate-length bias we propose is always less than the pitch of the layout grid; this avoids design rule violations. Moreover, it implies that the biased and unbiased cell layouts are completely pin-compatible and hence layout-swappable. This allows

biasing-based leakage optimization to be possible at any point in design flow unlike sizing-based methods.

- With a biasing of $8nm$ in a $130nm$ process, leakage reductions of 24% to 38% are achieved for the most commonly used cells with a delay penalty of under 10%.
- Using simple sizing techniques, we are able to achieve up to 33% leakage savings with less than 3% dynamic power overhead and no delay penalty. Use of more than two gate-lengths for the most commonly used cells along with improved sizing techniques is likely to yield better leakage savings.
- The devices with biased gate-length are *more* manufacturable and have a larger process margin than the nominal devices. Biasing does not require any extra process steps, unlike multiple-threshold based leakage optimization methods.
- *Lgate* biasing leads to more process-insensitive designs with respect to leakage current. Biased designs have up to 41% less leakage worst-case variability in the presence of inter-die variations as compared to nominal gate-length designs. In the presence of both inter- and intra-die CD variations, selective *Lgate* biasing can yield designs less sensitive to variations.

IV.D Standard-Cell Library Optimization for Leakage Reduction

Motivated by potential benefits of transistor-level biasing, in this section we discuss intelligent augmentation of the standard cell library. Our method eliminates the high runtime, and the limited design space constraints associated with previous techniques.

To further motivate the need for a transistor-level assignment scheme, Figure IV.5 shows the I_{off}/I_{on} curves for NMOS and PMOS devices for a range of small gate-length biases. For this 90 nm technology, the graphs indicate that the I_{off}/I_{on} vs. *Lgate* spread is much larger for PMOS devices. Clearly in this case, biasing PMOS devices provides larger leakage reduction for a given delay overhead. Therefore, an inverter with

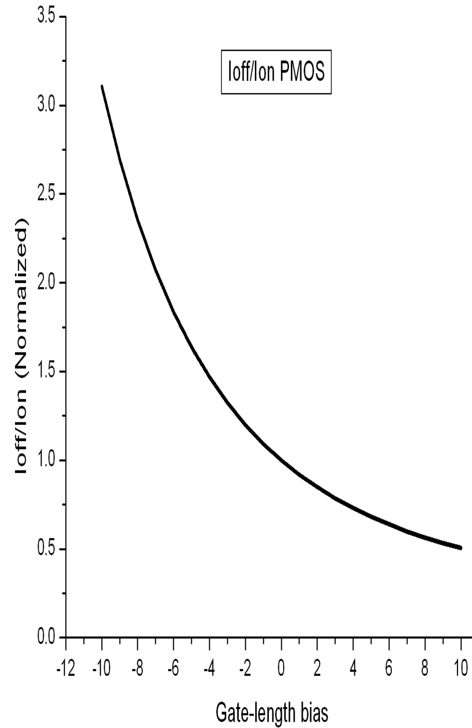


Figure IV.5: I_{off}/I_{on} characteristics for PMOS and NMOS devices.

higher bias assigned to PMOS will have a better leakage/delay tradeoff than one with equal biases assigned to both NMOS and PMOS devices.

From the graphs, it is also evident that with larger biases, the improvement in the I_{off}/I_{on} metric decreases. We conclude that smaller biases are more beneficial and limit ourselves to 10% of the nominal gate-length. Typically, process and area considerations constrain the bias even further and we limit ourselves to the bound imposed by the technology[10]. Taking these observations into consideration, we implement a cell-variant generation methodology in a module called Transistor-Level Biasing (TLB). Library optimization methods such as TLB are particularly attractive because the variant generation and characterization effort is amortized over multiple designs using the technology. Contributions of this work include:

[10]The constraints are imposed primarily by the polysilicon to polysilicon minimum spacing rules and polysilicon gate to active contact spacing rules.

1. taxonomization of various cell-variant classes; and
2. efficient algorithms to systematically augment a standard-cell library to drive design power optimization.

The rest of this section explains the variants as well as the generation algorithms in detail.

IV.D.1 Biasing Objectives and Cell-variants

Library optimization techniques are required to determine the best tradeoff between library size and design space. It is desired to give as large a design exploration space as possible to a design optimizer, while maintaining the size of the library within reasonable bounds. It is therefore, important to carefully determine the cells that would prove most useful in the circuit optimization process. The choice of variants is influenced by technology constraints, layout and design rule constraints, as well as by typical slack characteristics of designs.

To motivate the need for transistor-level biasing variants, we compute average slack per cell as well as average discrepancy between rise and fall slacks. Consider the timing report statistics for a few benchmark designs as shown in Table IV.10. The large discrepancy between rise and fall slacks is immediately obvious. In our sample set, this difference is found to be as large as 960 ps. A downstream power optimization engine will try to consume as much positive slack as possible to recover leakage power. These observations lead us to identify several useful biasing objectives. The variants corresponding to the identified objectives are described in Table IV.11[11]. We distinguish between cell-level biased (CLB) variants, where all devices have equal bias and transistor-level biased (TLB) variants.

CLB Variants

- Maximum Leakage Reduction: Cells on paths with large positive slack can be replaced with variants (C_Pmax) which have all devices biased to the maximum

[11]Although the current work focuses primarily on leakage reduction (_P variants), negatively biased variants are included for the sake of completeness. Ongoing work involves the use of _N variants for timing optimization. All test circuits in are initially timing-correct.

positive limit.

- Maximum Timing Improvement: Cells on paths with large negative slack must be replaced with cells (C_Nmax) which have all devices biased to maximum negative value.

We also (optionally) generate other CLB variants (C_Pn and C_Nn) where the biases are some fraction of the maximum bias, for paths with small positive or negative slack.

TLB Variants

- Leakage Reduction with Delay Upper Bound. Small positive slack can be exploited by TLB variants (A_P) that reduce leakage while maintaining the delay within a specified bound.

- Delay Reduction with Bound. When there is small negative slack, it is useful to have variants (A_N) with delay reduction that is some fraction of the maximum possible reduction, to avoid excessively large leakage overhead.

- Leakage Reduction with Transition-dependent Delay Overhead. The timing reports statistics suggest that some paths have large rise slack but little fall slack, and vice versa. For cells on such paths, it is very useful to have variants (R_P and F_P) that reduce leakage by slowing down one transition while keeping the other transition intact.

- Transition-dependent Delay Reduction. These variants (R_N, F_N) are for paths with negative slack for one transition and zero or positive slack for the other.

- Leakage and Delay Reduction. Finally, we propose special variants which we refer to as dominant (D) variants. Dominant variants are those that are superior in both delay and leakage to the nominal cell, or superior in one and equal in the other. These variants do not exist for all cells, and are possible only for technologies that allow both positive and negative biases. We motivate the existence of dominant variants by taking a simple example of an AND gate.

Table IV.10: Slack characteristics of circuit timing reports. All values are in ps.

Circuit	Avg. Slack	Max Slack	Avg.| R-F |	Max (R-F)	Max (F-R)
S38417	538	1540	68.7	300	300
AES	325	990	36.1	180	90
ALU	177	1370	21.1	40	100

Table IV.11: List of variants and polarity of biases.

Variant	Objective	Bias assignment
C_Pmax	Maximum leakage reduction	All Positive Max
C_Nmax	Maximum delay reduction	All Negative Max
C_Pn	Leakage Reduction: fraction of C_Pmax	Positive – Equal across devices
C_Nn	Delay Reduction:fraction of C_Nmax	Negative – Equal across devices
A_P	Leakage reduction. Delay upper bound	Positive.
A_N	Delay reduction with bound	Negative
R_P	Leakage reduction. Only fall delay affected	Positive
F_P	Leakage reduction. Only rise delay affected	Positive
R_N	Rise delay reduction	Negative
F_N	Fall delay reduction	Negative
D_P	Delay and Leakage Reduction. Emphasis on Leakage	Positive and Negative
D_N	Delay and Leakage Reduction. Emphasis on Delay	Positive and Negative

The circuit diagram for an AND gate is shown in Figure IV.6. Table IV.12 shows the state of each device in the circuit for different input states. The states are Delay Dominant (D), Leakage Dominant (L), Neither Delay nor Leakage Dominant (N).

A device is considered as delay dominant for a transition if it is in a charging/discharging path for that transition. It is considered as leakage dominant if it is turned off and not stacked (series connected to other *off* devices). We draw the concept of dominant states from [79].

From the table, M3, M4 and M5 contribute to the same transitions (transitions leading to input state '11'), while contributing very differently to average leakage (M5 leaks for three states while M3/M4 leak for only one). We can expect that an intelligent exchange of bias between M3/M4 and M5 (specifically, increasing the bias of M5 and

Table IV.12: Distribution of states over different devices in AND gate.

Input State		Device					
A	B	M1	M2	M3	M4	M5	M6
0	0	D	D	N	N	L	D
0	1	D	L	L	N	L	D
1	0	L	D	N	L	L	D
1	1	L	L	D	D	D	L

reducing that of M3/M4) can give us a variant with lower average leakage than the nominal cell, while maintaining similar delay characteristics.

As another example, we consider a cell where multiple input stages feed a single output stage. Assigning negative bias to devices in the output stage speeds up all transitions. The available slack can be used to reduce leakage by positively biasing devices in all input stages.

The examples suggest that dominant variants should be more common with multi-stage gates and this hypothesis is corroborated by experiment.

In the next subsection we describe methods of pruning the variant list under runtime/characterization constraints.

IV.D.2 Variant List Pruning

Due to runtime constraints for SPICE characterization of variants, as well as for optimization runs, it is sometimes required to prune the cell-variant list. We investigate the biasing benefits of different cells based on their usage statistics and topologies.

Cell Usage Statistics. One of the most obvious and useful ways of determining the number of variants to be assigned to every cell is observation of cell usage statistics over a few sample circuits. Heavily used cells should be assigned the largest number of variants. On the other hand, for cells that are very sparsely used or are not very leaky, the characterization and optimization runtime effort associated with a large number of variants would not be justified.

Topology. Heavily stacked devices usually have a small number of leakage dominating states, and their contribution to total cell leakage is small. It is not very useful to assign positive bias to these stacked devices, as there is considerable delay overhead for very small leakage gain.

This observation suggests that cells that have NAND topology (NMOS stacks) are not highly suited to R_P variants, as the biases are exclusively on the stacked devices. Similarly, cells with NOR topology are not suited to F_P variants. For inverters and buffers, the pull-up and pull-down networks are exactly the same. If for a particular

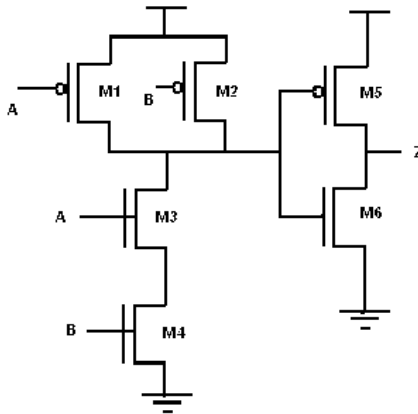


Figure IV.6: AND circuit diagram.

technology, PMOS and NMOS delay/leakage tradeoffs are very similar, A_P and A_N variants would not perform much better than C_Pn and C_Nn variants, and could, therefore, be left out. Also, for several cells, it is not possible to create dominant variants, and for some cells, where dominant variants are generated, the improvements are too small to justify the characterization overhead.

Having described all variants, in the next subsection we detail the biasing methodology used for generating the variants.

IV.D.3 Biasing Methodology

We now describe our heuristic for generating the variants described in Table IV.11. We first introduce an important concept, the biasability of a device.

Biasability computation

The biasability of a device is a figure of merit for assigning a larger bias to a device.

The basic definition of biasability for the leakage reduction objective is

$$B = \Delta Leak / \Delta Del \quad \text{(IV.1)}$$

This definition is modified slightly to account for the different objectives described previously, but the concept remains the same.

C_P and C_N variants do not require biasability computation as all devices are unconditionally pushed to the same bias. The definition of biasability in Equation (IV.1) is used for generating variants of A_P and A_N type. For A_P variants, this means that we can use the available slack in the most leakage-beneficial way as possible.

For the generation of R_P and F_P type variants, we use a modification of the biasability equation. The example given is for R variants; the F variants are analogous. Here,

$$B = \Delta Leak/(\Delta Del_{rise} + k) \tag{IV.2}$$

Here ΔDel_{rise} is the average delay overhead for all rise transitions. The constant k is chosen so as to render the biasabilities of devices that significantly affect the rise transitions nearly zero. These are primarily the devices that appear in a charging/discharging path during these transitions. Other transistors also affect the rise transition by appearing as a load in a charging/discharging path. These transistors have biasabilities much higher than the 'on' devices but lower than those that do not affect the transition at all. This ensures that all the devices that significantly affect the rise transitions are not biased, those that do not affect the transition are biased as high as possible, and the remaining devices have intermediate biases. R_N and F_N variants use this definition of biasability too.

For 'dominant' variants, we use the maximum delay overhead number in the denominator, as opposed to the average overhead.

$$B = \Delta Leak/\Delta Del_{\max} \tag{IV.3}$$

This is motivated by the fact that a 'truly dominant' variant is to be used to replace the existing cell in the library, and should therefore be superior in leakage as well as *per-arc* timing to the unbiased cell.

An optional practical consideration in this step is the handling of fingered devices. Typically, each finger is treated as an individual device by TLB. However, physical verification tools merge fingers into a single device for runtime considerations. This functionality is hampered by assigning different gate-lengths to each finger. Under such

```
     Algorithm: generateTLB
1.   computeBiasability();
2.   For all i, bias[i] = minBias;
     x=0;
3.   Iterate:
         x=x+1 ;
         bias[i] = x×biasability[i];
        Snap bias to grid;
        computeDelayOverhead();
4.   If Delay overhead > Delay Upperbound
         solution = previous bias;
         return solution;
     else goto Iterate;
```

Figure IV.7: Basic biasing algorithm.

conditions, TLB can force the biasabilities of all fingers to be equal, therefore ensuring that they are assigned equal biases.

Biasing algorithm

In this section, we describe the variant generation algorithm. The core algorithm is as described in Figure IV.7. We note that in Step 3, we snap the biases to a pre-defined grid at every iteration. The grid is defined by technology parameters. The changes to the algorithm from Figure IV.7 for different variants are outlined below.

1. *A_P*: Identical to Figure IV.7.
2. *A_N*: *minBias* is set to maximum allowed negative bias. Exit condition is failing to meet the required delay improvement.
3. *R_P (F_P):* Exit condition changed to whenever it is found that all the primary rise (fall) transition-affecting devices have reached their maximum bias values.
4. *R_N (F_N)*: Similar to *A_N* with exit condition being all devices that do not affect rise (fall) transitions (either directly or through loading) are ‘unbiased’.
5. *D*: Similar to *A_N* with exit condition as finding a biasing solution that has lower leakage than the nominal cell. A *D* variant is found if the variant delay is less than

Table IV.13: Average delay and leakage overheads for all variants.

Variant	Rise Delay Overhead (%)	Fall Delay Overhead (%)	Leakage Overhead (%)
A_P	7.42	4.48	-31.92
A_N	-4.76	-4.57	34.81
R_P	-1.32	10.84	-17.67
F_P	9.11	-0.98	-23.60
R_N	-6.20	-1.80	60.97
F_N	-1.06	-6.24	89.88
D	-0.62	-0.99	-3.23
C_P4	5.77	4.66	-26.73
C_N4	-5.39	-3.97	36.79
C_P6	8.26	6.71	-36.27
C_N6	-7.99	-5.98	54.76

nominal.

Table IV.13 shows the average delay/leakage tradeoffs for the variants observed after characterization. C_P4 and C_P6 indicate cells where all devices are biased at 4nm and 6 nm respectively. The R_P and F_P variants have delay overheads only in the F and R transitions respectively. Similarly, the R and F_N variants have only the R and F transitions sped up significantly. The D variants have small improvements but they are strictly better in delay as well as leakage.

The A variants[12] clearly show the tradeoff improvements achieved by using TLB over CLB. We compare the A_P variants with the C_P variants using the $\Delta Leak/\Delta Del(avg)$ metric. The value of this metric for A_P variant is 5.36, while for C_P4 it is 5.13 and C_P6 it is 4.84.

Similarly, we compare the A_N metric with C_N4 and C_N6 through the $\Delta Del(avg)/\Delta Leak$ metric. The value is 0.134 for A_N, while it is 0.127 for both C_N4 and C_N6. Clearly, the TLB variants have a more favorable bias assignment compared to CLB variants, for both slack utilization and timing optimization.

[12]Here the A variants are generated such that the delay change for these variants is 75% of the maximally biased C variants.

IV.D.4 Delay/Leakage Models

For the algorithms in the previous subsection, we need to compute delay and leakage overheads at every biasing step. Since several iterations are required to reach the final bias value, the delay and leakage computation models should be fast as well as accurate enough to reach an acceptable solution. We implement a transistor-level delay model (TLM) similar to [117].

Delay Modeling

At the core of our delay modeling routine is an RC delay model The delay is recomputed for every target input state. Currently the model does not distinguish between different input transitions leading to the same output state.

A set of channel connected devices is referred to as a stage, and forms the basic unit of analysis. The modeling routine is explained with the help of the two-stage AND gate described previously. Using Table IV.12, we determine the delay-dominant devices corresponding to each input state. Performing series-parallel reduction on the dominant devices, each stage is reduced to an RC pair. Both gate and junction capacitances are considered. As an example, for transitions leading to input state '11', the delay is expressed as

$$D = (R3+R4)(C5+C6+CJ1) + R5\times(CL+CJ2)$$

Here Ri and Ci are, respectively, the resistance and capacitance of Device i and CJi is the junction capacitance of Stage i, determined by weighing each device junction capacitance by its corresponding resistance.

The resistance of each device is a function of its gate-width and length. TLM obtains these values from look-up tables generated by SPICE pre-characterization.

Leakage Modeling

To estimate the leakage of the cell, once again we refer to Table IV.12. The leakage-dominant states are determined as described earlier in this section. The total cell leakage corresponding to a state is simply the sum of the off-currents of the dominant devices. Similar to resistance values, the off-currents are obtained from a look-up table generated by SPICE pre-characterization.

The accuracy of these models suffers due to layout-dependent effects such as well proximity, stress, etc. as well as the "lumped" nature of the delay model we use in the current implementation. However, we note that the absolute delay and leakage values are not of interest here and only the relative overheads due to biasing are required to be accurate. As is shown by the characterization results in Table IV.13 and optimization results discussed in the next subsection, the level of accuracy provided by TLM is sufficient for optimization purposes.

IV.D.5 Optimization setup and Results

For our tests, we use an industrial 90 nm technology with BSIM 4.3 SPICE models. The optimization is carried out by a sensitivity based optimizer similar to [61]. For correctly using TLB variants a slack-aware sensitivity function is essential. This enables the optimizer to choose the appropriate variant for the particular value of available slack. This is especially important for R/F variants, where a slack-unaware sensitivity function may incorrectly prefer a variant with lower average delay overhead ignoring any transition-dependent slack discrepancy.

We test our implementation on designs from the ISCAS-89 suite [39], the Opencores [163] suite. Optimization results are shown in Table IV.14. The tests were carried out on three libraries as follows.

1. CLB-only library containing only CLB variants with various bias values.
2. CLB+TLB library with some of the CLB variants replaced with TLB variants while maintaining the same library size.
3. Complete library with all available variants.

In the first two cases, the number of variants and hence the library size was maintained to be the same. Results show that new libraries achieve significant leakage reduction over the existing design. Also, comparisons between CLB and TLB libraries clearly demonstrate the superior slack utilization capabilities of various TLB variants. Since the library size is the same, the runtime of both TLB and CLB-based optimization is nearly the same. Library 3 has a larger number of variants, improving the leakage

Table IV.14: Optimization results for TLB and CLB based libraries.

Circuit	Instance Count	% Imp. CLB	% Imp. TLB	% Imp.All Variants
C5315	1681	27.66	41.69	42.09
C6288	3041	16.99	26.17	27.25
AES	30991	22.68	38.05	38.66
ALU	15880	15.68	32.56	33.13
S9234	1212	24.38	31.41	32.46
S13207	3464	30.83	40.15	40.43
S38417	11620	25.98	38.44	38.78

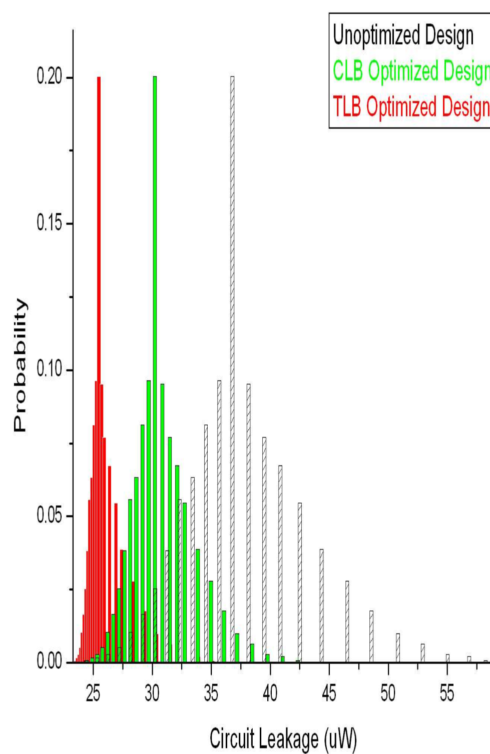


Figure IV.8: Pre and post optimization leakage distribution for AES.

reduction slightly, while increasing total runtime. The results show that we achieve, on an average, 36% leakage improvement over unoptimized designs and 12% leakage improvement over CLB-optimized designs. Going back to Figure IV.5, we note that increasing the mean of the gate-length moves the bias point to the 'flatter' region of the gate-length/leakage curve. Therefore, apart from reducing the mean of the leakage, we also expect to reduce its variability with respect to gate-length fluctuations. In other words, the standard deviation of the leakage distribution is reduced. The plots in Figure IV.8 show the distribution of gate-leakage obtained by Monte-Carlo simulation for unoptimized, CLB-optimized and TLB-optimized designs for the AES benchmark. The distribution for the TLB optimized circuit is not only shifted to the left considerably, it is also much tighter compared to the other designs. Here, the standard deviation is 66% less than the unoptimized design and 39% less than the CLB-optimized design. The increasing power-limited yield loss in scaled technologies makes this reduced sensitivity to linewidth variation a highly desirable characteristic.

IV.D.6 Conclusions

In this part of the chapter, we have proposed a new standard-cell library optimization method for leakage reduction. Existing standard cells are modified by performing transistor-level gate-length biasing, to change their leakage-delay characteristics. The enhanced library thus generated is used by a power optimizer to generate a design with lower leakage power consumption than the original design. This method also considerably reduces the sensitivity of leakage to gate-length variation. Overall, we obtain leakage reduction of up to 42% and leakage variability reduction of 66% by applying our algorithm to an unoptimized design. Compared to a design optimized with only cell-level biased variants, we achieve up to 17% additional reduction in the mean and up to 39% reduction in the standard deviation of leakage with no runtime overhead.
Ongoing work on this project is primarily in the following areas:

1. Use of negatively biased variants for timing optimization as well as for enhanced leakage optimization using hill-climbing algorithms.

2. Improvement of delay/leakage modeling accuracy.

3. Added variant generation flexibility by incorporating threshold voltage assignment.

4. Variant generation specifically targeted for sequential cells.

IV.E Acknowledgments

Chapter IV is in part a reprint of "Gate-Length Biasing for Runtime Leakage Control", *IEEE Transactions on Computer-Aided Design*, 25(8) (2006) and "Standard-Cell Library Optimization for Leakage Reduction", *Proc. ACM/IEEE Design Automation Conference*, 2006. I would like to thank my coauthors Puneet Sharma, Saumil Shah, Dr. Dennis Sylvester and Dr. Andrew B. Kahng.

V

The Loop Back from Lithography Simulation

V.A Lithography Simulation-Based Full-Chip Design Analyses

RETs are key enablers of the aggressive IC technology scaling that has fast outpaced advancements in lithography hardware solutions. RETs such as optical proximity correction (OPC), phase-shift masks (PSM), and off-axis illumination (OAI) dramatically improve resolution and are extremely effective at process variation control. The increased mask and manufacturing costs due to the application of these techniques have been outweighed by the advantages offered, and these techniques are imperative during mask-data preparation. RETs modify the design significantly and there is little similarity left with the design at the post-layout stage at which sign-off is performed. At non-ideal process conditions significant process variations can result even within the process window. Due to RETs and process variations features do not print at their nominal dimensions causing circuit power and performance to be significantly different from sign-off estimates. Today's design flows worst-case process effects and consequently overdesign circuits leaving valuable performance on the table.

Lithography simulation[1] enables estimation of CD variations at different process

[1]Residual OPC critical dimension (CD) error or post-OPC edge placement error (EPE) results can be easily used to generate an output similar to that of lithography simulation by adding the errors to

points. According to the international technology roadmap for semiconductors (ITRS), a substantial fraction of variations is systematic and can be modeled accurately after layout [118]. So even though random variations cause differences between on-silicon shapes and those predicted by lithography simulation, these difference are relatively small. Consequently, lithography simulation-based design analyses are likely to be significantly more accurate to on-silicon than post-layout analyses. Current lithography simulation tools are completely shape-based and not connected to the design in any way. We present a novel methodology that uses the results of lithography simulation for estimation of performance and power of a design using standard device- and chip-level analysis tools. The proposed approach can reduce worst-casing and can facilitate optimizations that account for process variations.

A recently proposed work by Yang et al. addressed post-lithography based analysis and optimization. They proposed a timing analysis flow based on residual OPC errors (equivalent to lithography simulation output) [111]. In this work lithography simulation was performed only on timing-critical cells and their neighborhood. In modern designs a large fraction of cells are timing-critical and along with their neighborhood can include almost all cells limiting the runtime benefits of the approach. Only setup-time analysis was performed and interconnect variations ignored. Several non-trivial details related to handling non-rectangular gates in SPICE simulations and cell-level hierarchy reconstruction are missing. Recent works have also attempted to capture systematic variations and account for them in analyses and optimizations. Gate-length variability due to proximity effects and across-field lens aberrations was characterized by Orshansky et al. with a set of simple patterns located at different field locations [121]. Systematic variations due to defocus and pitch were captured through lithography simulations of simple test-patterns and used to drive timing and leakage analyses and optimizations [104, 105]. Systematic variability due to lens aberrations was characterized for timing analysis and analytical placement [120]. These works, however, rely on the ability of simple patterns to predict process variations which unfortunately can be quite inaccurate especially as the optical radius of influence (i.e., radius beyond which proximity effects fade) increases with technology scaling.

---

the drawn CD.

An overview of our approach is as follows. For MOS devices, performance and power is dependent on their gate-length and gate-width. To compute the gate-width, the active region contour is approximated by an equivalent rectangular region [91]. Accurate determination of gate-length is more important due to the heavy dependence of leakage and delay on it. Our gate-length computation depends on the objective to be analyzed (e.g., delay, leakage, dynamic power). We first rectilinearize the simulated gate contour and then approximate it by an electrically-equivalent rectangle. We then use transistor-level modeling (TLM) to estimate the impact of gate-length variations on design metrics. As an alternative to TLM, we propose a cell-level analysis flow that allows standard analyses tools to be used. After lithography simulation cell instances of the same cell differ and cannot be mapped to the same cell in the library for lithography simulation-based analyses. We add variants of each cell in the library; the variants are similar in function and drive strength of the cell but have different gate-lengths assigned to the devices. After rectilinearization and determination of gate-length of all devices in a cell instance, the variant that matches in the electrical behavior of the cell instance is selected and mapped to. The output is generated in the form of a modified Verilog file and can be used by standard analyses tools. The above-mentioned flow generates separate Verilog files for different metrics to be analyzed. We propose a "mixed-mode" flow in which only one Verilog file that is accurate for all metrics is generated. Our interconnect analysis flow modifies the parasitics database to account for variations in wire width and spacing. Interconnects are simplified to polygons and their resistance computed using analytical formulas. For capacitance computation, pairs of interconnects are simultaneously simplified and the change in their coupling capacitance estimated using a pre-created look-up table. The same parasitic extraction approach is used to compute parasitics for corresponding drawn shapes and the *change* in parasitics is computed. The parasitic database is then updated with the change.

V.A.1 Device Analyses

The gate-length and gate-width of a MOS device have the most direct impact on its performance and power. While performance and power exhibit complicated dependence on gate-length and gate-width, simpler approximations are as follows: (a) delay

is partially determined by the saturation current which increases and decreases linearly with gate-length and gate-width respectively, (b) dynamic power increases linearly with gate-length and gate-width due to the change in gate capacitance, and (c) subthreshold leakage increases linearly with gate-width and exponentially with the squared of gate-length. Our analyses accounts for changes in gate-length and gate-width due to lithography imperfections and those in associated parameters such as source area and parameter, drain area and parameter, and stress parameters. Consequent changes in parasitics, however, are ignored in our analyses.

Device Gate-Length and Gate-Width Computation

The gate is formed where the poly and active regions overlap. The non-rectangular shapes of poly and active regions makes the gate-width computation non-trivial. We use the flow previously proposed to find the *equivalent* active region and compute gate-width (W_{Avg}) [91].

Delay and power are heavily dependent on the small variations in gate-length introduced by lithography. Therefore, it is important to accurately access the delay and power impact due to variations in gate-length. After a sequence of simplification steps described below, a gate contour is reduced to a rectangle and the *average* gate-length computed. We differentiate between the different analyses metrics - setup time, hold time, leakage, and dynamic power - for simplifications to preserve electrical equivalence as much as possible. The first step is to rectilinearize the gate contour generated by lithography simulation. Three possible ways for rectilinearization are: (1) interior-point, (2) exterior-point, and (3) mid-point and are illustrated in Figure V.1. For setup time as the objective, exterior point rectilinearization may be performed. This is because gate delay, transition time, and capacitance increase with gate-length; exterior-point rectilinearization yields an upper-bound on the gate-length and consequently setup time. For similar reasons interior-point rectilinearization may be used for hold time and leakage. Dynamic power depends on the gate capacitance which is determined by the gate area. Therefore, an area-preserving rectilinearization is desirable for dynamic power; mid-point rectilinearization may be used as a less computation-intensive approximation. While the described objective-specific methods of rectilinearization ensure pessimistic

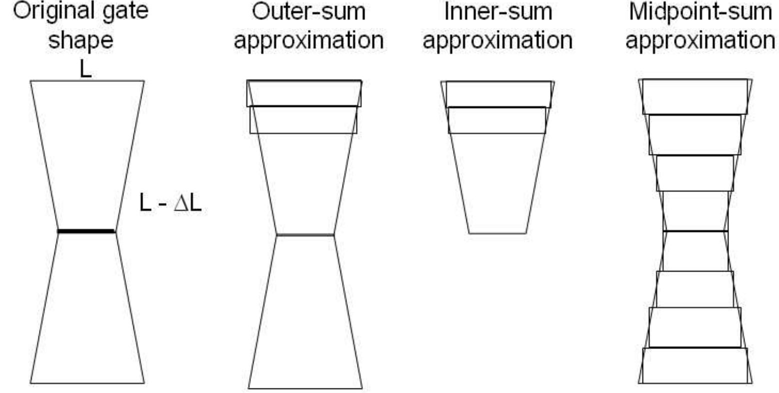


Figure V.1: Three possible ways for rectilinearization.

estimates, we use only mid-point rectilinearization in our experiments to reduce memory requirements.

Since standard circuit simulation tools such as Synopsys HSPICE can only simulate rectangular gates, the next simplification step reduces a rectilinear polygon to a rectangle. L_{avg} denotes the gate-length of the rectangle and is computed differently for the various analyses objectives. We allow two modes for L_{avg} computation:

1. *Look-up table mode.* In this mode we use a flow similar to one proposed previously [91]. Look-up tables for device on- and off-currents are created for different gate-lengths and gate-widths by SPICE simulations. The rectilinear polygons are sliced into rectangles and the on-currents (for setup and hold delays) and off-currents (for leakage) of all slices are summed up. L_{Avg} is the gate-length of the rectangle of the same gate-width and that yields the same on- or off-current. For dynamic power objective, rectangle that preserves the area is used.

2. *Expression mode.* If device SPICE models are not available, the look-up tables cannot be generated. In this case analytical expressions may be used to compute L_{avg} for different objectives. For example for dynamic power objective an area-preserving rectangle would have $L_{avg} = \sum_i W_i L_i / W_{avg}$, where W_i and L_i are the width and length of the ith slice.

Cell-Level Analyses.

While device-level analyses tools are more accurate, they are not sufficiently fast to be run full-chip. Standard full-chip analyses tools are cell-level; they use characterized libraries that are created by SPICE simulations to perform design analyses. We propose two flows to use standard-cell-level analyses tools for lithography simulation-based analyses:

1. *Cell library-based*. Cell-level performance and power analyses tools require each cell *instance* in the design to refer to a cell *master* in the library. All drawn cells that refer to one cell master are alike. However, after lithography simulation they may all differ. Unfortunately, it is not feasible to create different cells masters for each instance and the hierarchy needs to be reconstructed. We create a library that contains multiple variants of each cell master. Two types of library variants may be characterized: (1) *cell-level* in which gate-lengths of devices in a variant are equal but differ from those of devices in another variant, and (2) *transistor-level* in which devices in a variant may have different gate-lengths. To reduce library size, we do not alter device gate-widths in the variants because the percentage variation in gate-width after lithography simulation is very small. For each cell instance, a different cell variant may be chosen for different objectives. For setup time, the variants in which gate-length of each device is larger than the L_{avg} of the corresponding device are selected. Similarly, for hold time and leakage objectives, the variants in which gate-length of each device is smaller than the L_{avg} of the corresponding device get selected. For dynamic power, the variants in which the gate-area of each device is larger than the gate-area of the corresponding litho-simulated device are selected. When multiple variants meet the selection criteria, the one that minimizes the error ($= |L^j_{avg} - L^j_{var}|$, where L^j_{avg} is the L_{avg} of device j and L^j_{var} is the gate-length of device j in the variant) is chosen.

   After a cell variant is chosen for a cell instance for an analyses objective, the Verilog file is modified to reflect the binding of the cell instance to the cell variant. The modified Verilog can be used by an off-the-shelf analyses tool. The accuracy of the analyses increases with the number of variants in the library at the cost of library

characterization time.

2. *Transistor-level modeling.* If library characterization is not feasible, transistor-level modeling (TLM) may be used to estimate the variations in performance and power due to process variations. At the core of our delay modeling routine is an RC delay model. The delay is recomputed for every target output state. Currently the model does not distinguish between different input transitions leading to the same output state. A set of channel connected devices is referred to as a stage, and forms the basic unit of analysis. The next step is to identify devices that contribute significantly to a particular transition. These dominant devices are all devices that are part of a stack that connects the stage output to power or ground. Performing series-parallel reduction on the dominant devices, each stage is reduced to an RC pair. Both gate and junction capacitances are considered. The delay of each stage is expressed as the product of the equivalent resistance and the load capacitance of that stage. The resistance of each device is a function of its gate-width and length. TLM obtains these values from pre-created look-up tables generated by SPICE simulations. The computed delay overheads are used to modify the timing arc delays during static timing analysis to facilitate full-chip timing analyses.

   To estimate the leakage of the cell, we identify leakage dominant devices for each input state. It is known that stacked devices have very low leakage. Only devices that are in the off state and are not series connected to any other off devices are labeled dominant. The total cell leakage corresponding to a state is simply the sμm of the off-currents of the dominant devices. Similar to resistance values, the off-currents are obtained from a look-up table generated by SPICE simulations. The average leakage of the cell is the average over all input states. We note that the absolute delay and leakage values are not of interest here and only the relative overheads due to gate-length variation are required to be accurate. Table V.1 shows the delay and leakage overhead accuracies for our transistor-level modeling method for various cells, for a particular length assignment. The accuracy suffers due to layout dependent effects such as well proximity, stress, etc. as well as the "lumped" nature of the delay model we use in the current implementation. However, the level

Table V.1: Transistor-level modeling matching accuracy.

Cell	Delay Overhead (%)		Leakage Overhead (%)	
	SPICE	TLM	SPICE	TLM
INV	-4.8	-8.1	42.68	49.37
NAND	-7.3	-11.2	53.93	60.69
AND	-6.8	-7.8	50.02	56.56
AOI	-6.3	-6.5	51.82	57.49
MUX	-5.8	-4.7	50.94	56.77

of accuracy provided by TLM is sufficient for the modeling purpose described in this chapter.

V.A.2 Interconnect Analyses

Our interconnect analysis flow computes the change in parasitics caused due to mismatch between drawn and litho-simulated interconnect shapes. We update the standard parasitic extended format (SPEF) database with the changed parasitics and then an off-the-shelf timing analysis tool can be run. If the output of lithography simulation are contours, we simplify the contours to polygons by a piecewise linear approximation. Resistance of an interconnect is computed by integration over the length. Since interconnects are polygons this reduces to addition of resistances of trapezoids (from top view) and can be done very efficiently by analytical formulas.

We iterate over coupling capacitances found in the SPEF database and analyze the two interconnects between which the coupling capacitance is computed simultaneously. Since SPEF may have long interconnects fractured during parasitic extraction, we use node coordinates, that can be optionally specified in SPEF, to establish a mapping between fractured interconnects and routing segments in the design. Figure V.2 shows the steps involved in our shape simplification flow for coupling capacitance computation. Without loss of generality we assume the pair of interconnects to be vertical. Horizontal lines are drawn through all vertices of the two interconnect polygons to cut the polygons into sections. Two adjacent horizontal lines contain a section pair, one from each of the two polygons, between them. To compute the coupling capacitance between a section pair, the sections are split into horizontally-aligned micropanels. Values of capacitances

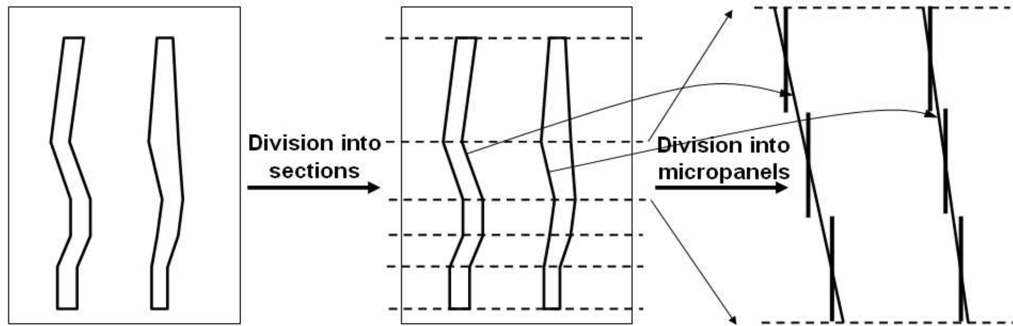


Figure V.2: Steps involved in shape simplification for capacitance computation.

for a pair of horizontally-aligned micropanels are obtained from a look-up table. The look-up table lists values of parallel plate capacitance and also includes fringing components of the capacitance values, which are impacted by the presence of metal wires above and below pair of polygon shaped metal wires. Capacitances for different micropanels are summed up to find the capacitance between sections and summed up for all sections to find capacitance between the interconnect pair. With the same method, we also find the capacitance for corresponding drawn shapes and compute the change in capacitance to update the SPEF database. The capacitance look-up table is created using 3D field solver simulations for template geometries generated for a technology. Capacitance values are obtained after interpolation using the following parameters: (1) widths of the interconnect pair, (2) spacing, (3) layer, and (4) densities of above and below layers. The runtime increases with the complexity of the interconnect polygons and the number of micropanels created.

V.A.3 Full-Chip Analyses

Figure V.3 illustrates the complete analyses flow. The DEF file, that contains layout information for cells and interconnects along with connectivity, is used to correlate the shapes in litho-simulated GDS with cells and interconnects. Within a cell, device locations are correlated with device names as a byproduct of layout versus schematic (LVS) between cell GDS's and SPICE netlists. Our full-chip analyses flow iterates over all cell instances and invokes device analyses. To improve the runtime, full-chip analyses optionally takes the optical radius (i.e., radius beyond which proximity effects fade) and

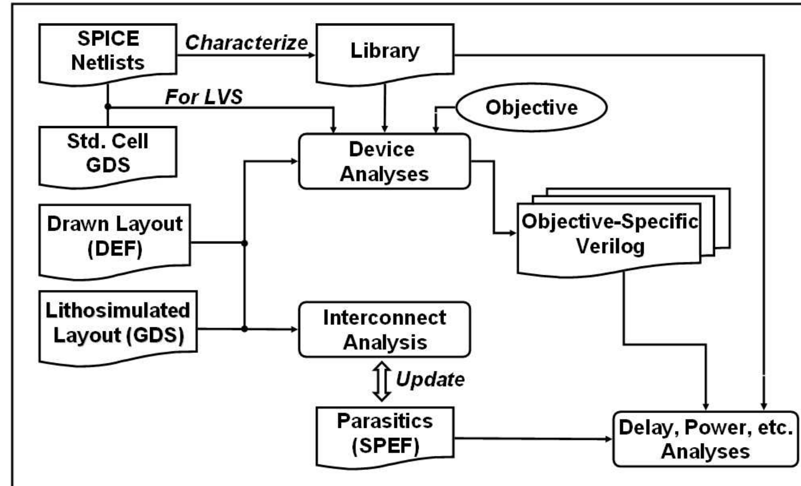


Figure V.3: Lithography simulation-based design analyses flow.

for cells that do not have other cells within the optical radius cached results are used instead of device analyses.

Mixed-mode analyses

As described in Section V.A.1, Verilog files differ for the various analyses objectives. Maintaining objective-specific Verilog files can be cumbersome and is non-standard. We propose a hybrid approach that assigns different objectives to individual cells, based on the sensitivity of an objective to each cell, to generate a single Verilog file that allows accurate analyses for all objectives. An objective for each cell is selected in the following order:

1. Hold time. We first perform hold-time analysis with objective set as hold. We note that hold-time analysis yields the most accurate hold estimates while setup-time analysis yields the most inaccurate hold estimates. So we perform hold-time analysis again but with setup as the objective. For each cell we find the difference between hold-time slack at the two objectives and if that difference is larger than a user-configurable fraction of the hold-time (for hold objective), then the cell is flagged as hold critical. Hold objective is assigned to the hold-critical cells. The user-configurable fraction allows near-critical paths and not just the most critical

Table V.2: Testcases used in our experiments.

Circuit	Source	Cells	Nets	IO Pads
s1423	ISCAS'89	1406	708	23
c5315	ISCAS'85	1520	1698	301
AES	opencores.org	25824	26083	388
OpenRisc	opencores.org	58999	59374	374

path to be flagged as timing critical.

2. Setup time. In a way similar to hold time, setup-critical cells are identified and the ones that are not hold-critical are assigned the setup objective. If many cells are found to be simultaneously hold- and setup-critical, mixed-mode analyses should not be performed.

3. Capacitance. To accurately load the hold- and setup-critical cells, capacitance objective is assigned to the cells that: (1) load cells with hold or setup objective assigned, and (2) have not been assigned an objective.

4. Leakage or dynamic power. Only one of leakage or dynamic power can be performed at a time. We assign the leakage or dynamic power objective to all cells that do not have an objective assigned.

V.A.4 Experiments and Results

In this section we present our experimental setup and results. We show that delay and power estimates after layout and from our flow differ considerably. The proposed flow is fast enough to be run on large testcases in practical runtimes.

Experimental Setup

Testcases used in our experiments are summarized in Table V.2. We use SPICE models and cell netlists from a leading foundry and commercial tools for cell characterization, and testcase synthesis, layout and extraction. We use Mentor Calibre v.v9.3_5.9 for OPC and lithography simulation.

Results

Table V.3 presents circuit delay, leakage, and dynamic power of our four test-cases analyzed: (1) after layout, (2) using the proposed flow with lithography simulation performed with zero defocus, and (3) using the proposed flow with lithography simulation performed with $100nm$ defocus. We observe that circuit performance and power is close to post-layout estimates at zero defocus. Unfortunately, RETs are not as effective at non-ideal process conditions and we observe significant change in performance and power are $100nm$ defocus. With our OPC recipes, linewidths tend to decrease with increasing defocus for most patterns. Therefore, leakage increases tremendously, circuit time improves and hold-time violations become likely.

Table V.4 presents the accuracy of mixed-mode analyses with respect to objective-specific analyses. As discussed in Section V.A.3, mixed-mode analyses generates a single design that is accurate for all objectives (i.e., matches for a particular objective with the design generated by analyses specific for that objective). We observe that setup slack, hold slack, and leakage estimates from mixed-mode analysis match reasonably well with analyses with setup, hold, and leakage as objectives respectively.

V.A.5 Conclusions

Power and performance estimates after layout can be substantially different from on-silicon performance. Lithography simulation predicts on-silicon geometries for given process settings. In this section we proposed a flow to use the lithography simulation results to predict on-silicon power and performance. For device analyses, we perform steps to simplify gate contours from lithography simulation to regular rectangular gates. To facilitate cell-level power and performance analyses, we proposed a methodology to map printed cells to cell variants (cells of similar functionality and drive strength but with non-nominal gate-lengths) in the library. We also proposed an alternative transistor-level modeling-based analyses flow. Imperfections in printing of the interconnects alter their parasitics and consequently performance. Our interconnect analyses flow simplifies the contours from lithography simulation and uses a pre-created look-up table to estimate the changes in parasitics. The parasitic database is then updated to enable lithography simulation-based timing analyses.

Table V.3: Delay, leakage, and dynamic power estimates after layout, and after lithography simulation at $0nm$ and $100nm$ defocus using the proposed flow.

Circuit	Post-layout			Litho-sim at 0nm defocus			Litho-sim at 100 nm defocus			CPU
	Delay	Leakage	Dynamic	Delay	Leakage	Dynamic	Delay	Leakage	Dynamic	
	(ns)	(μW)	(mW)	(ns)	(μW)	(mW)	(ns)	(μW)	(mW)	(s)
s1423	1.221	29.112	0.222	1.229	29.122	0.222	1.064	44.723	0.218	60
c5315	0.639	95.220	1.369	0.647	95.235	1.372	0.550	160.805	1.337	151
AES	2.155	582.707	2.858	2.157	584.917	2.864	2.135	904.541	2.797	2221
OpenRisc	0.700	3415.424	16.920	0.704	3411.665	16.962	-	-	-	5022

Table V.4: Accuracy of mixed-mode analyses with respect to individual objective-specific analyses. Circuit c5315 is combinational so hold-time analysis is not applicable.

Setup timing slack (ns)				
Circuit	Objective			
	setup	hold	leakage	mixed-mode
s1423	**-0.0043**	0.0000	0.0000	**-0.0039**
c5315	**0.0000**	0.0016	0.0016	**-0.0001**
AES	**0.1000**	0.1017	0.1017	**0.10173**

Hold timing slack (ns)				
Circuit	Objective			
	setup	hold	leakage	mixed-mode
s1423	0.0931	**0.0927**	0.0927	**0.0927**
c5315	NA	NA	NA	NA
AES	0.0002	**0.0002**	0.0002	**0.0002**

Leakage (μW)				
Circuit	Objective			
	setup	hold	leakage	mixed-mode
s1423	43.112	44.741	**44.723**	**44.616**
c5315	158.807	160.851	**160.805**	**160.771**
AES	895.552	904.751	**904.541**	**904.541**

In addition to speed and accuracy improvements of the various modules, we are specifically working in the following directions:

- Currently for L_{avg} computation we do not consider the impact of the slice locations. However, due to narrow-width effects, slices near the gate edge affect device performance and power differently than those at the center [119]. We plan to consider these effects for more accurate L_{avg} computation.
- For cell-level analyses we plan to develop a hybrid methodology that uses transistor-level modeling to interpolate between characterized cell variants to improve the analyses quality without the need for large number of variants.

V.B Modeling of Non-Uniform Device Geometries for Post-Lithography Circuit Analysis

Due to Line Edge Roughness and other effects, irregularities are observed in printed shapes. A silicon image of a transistor gate is often not confined to a perfect rectangle, as is assumed by all current circuit analysis tools. These tools are unable to handle complicated geometries. Large discrepancies are observed between the simulated and observed values of circuit parameters such as current and threshold voltage.

There have been several previous approaches to modeling non-rectilinear geometries [89, 90, 91]. A significant drawback in all these works is that they consider the threshold voltage and hence the current density to be uniform along the device width. As a result, variations in length are treated the same, irrespective of the location of the variation along the channel.

It is known that the fringing capacitance due to line-end extension, dopant scattering due to STI edges [94], and the well-proximity effect (WPE) [96, 97] significantly affect the device threshold voltage. These effects are more pronounced near the device edges and roll off sharply as we move towards the center of the device. The V_{th} and current characteristics are, therefore, different along the channel. The extent of the 'edge effect' is different for different technologies. Figure V.8 shows a TCAD simulation of subthreshold current density as a function of location along the device width for a particular gate-length. For this technology, it is observed that the edge current is up to 2x larger than the center current. It is important to note that a given variation in length near the edges has a significantly different impact on the current than a variation near the center of the device. In other words, to model the effect of non-rectilinearity on gate characteristics, it is important to consider not only the dimensions of the variation, but also its location along the device width.

In this section, we model the threshold voltage and hence the on and off current densities as functions of distance from the device edge. The coefficients of the model are empirically adjusted for different gate-lengths. We use this model to analyze a non-rectilinear gate by treating it as a composition of several small rectangles of different lengths. For each rectangle, we use the current density model corresponding to its

length. The total current of this rectangle is the integral of the current density over its width. The limits of the integral for a particular rectangle depend on its location, thus the value of the integral is different for each rectangle. The sμm of the currents of all rectangles is the total current of the device. The total current can be used to provide an equivalent rectangular length for the device, so that it can be modeled using SPICE like tools.

V.B.1 Physical Explanation of the Edge Effect

Figure V.4 shows the cross-section of a MOS device with Shallow Trench Isolation (STI) technology. To compromise the pull-back effect (i.e., line end shortening) due to defocus, certain amount of gate-poly should extend over diffusion area. The extension over the STI region is defined as line-end extension.

Though the deep buried well technique has several advantages over current IC design, which include the possibility of triple-well structures and providing a low resistance path to ground for SER reduction [97], these deep buried well layers have non-negligible impact on the devices adjacent to the mask edge. Variation in the threshold voltage for those devices is observed since there is high probability for scattered ions to be implanted in the silicon surface. This effect is called the Well Proximity Effect (WPE).

All these factors contribute to the Narrow Width Effect (NWE), which is dependent on the isolation process. For non-recessed and semi-recessed (e.g., LOCOS) isolation processes, one can expect that the V_{th} will increase as device width decreases [93, 94, 98]. On the other hand, Reverse Narrow Width Effect (RNWE) in which the V_{th} decreases as the device width shrinks is observed in STI process [93, 94, 98]. The RNWE can be explained in part by the parasitic fringing capacitances (shown in Figure V.4) between gate, STI sidewall, and active area, where there is poly gate extension over the isolation area.

V.B.2 Development of Location-Dependent V_{th} Model

The solid line in Figure V.5 shows I_{off} vs. Width graphs for an NMOS device for the 90 nm technology under consideration. These values are generated using a

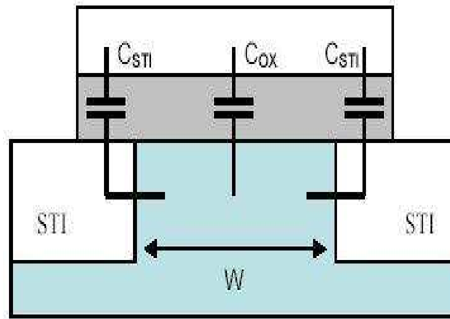


Figure V.4: Cross-section of device showing STI edges and fringing capacitances.

TCAD setup tuned to match an industrial 90 nm process closely. The device models corresponding to this industrial process are level 54 SPICE models.

The figure shows that, beyond a certain width, the current varies linearly with width, suggesting a nearly constant threshold voltage. However, as the width is decreased, the graph becomes increasingly flat till a point is reached where I_{off} increases with decreasing width, which we refer to as the current 'kick-back' effect. On the other hand, the I_{on} curve shown in Figure V.6 is nearly linear. These observations can be explained by modeling the threshold voltage as a piecewise function of location as shown in Figure V.7. The threshold voltage is constant at the center and decreases near the edges. At low widths, the effects from the two edges interact to reduce the V_{th} even further as shown in Figure V.7. I_{off} being an exponential function of V_{th}, the decrease in V_{th} due to width more than offsets the linear decrease in current. On the other hand, I_{on} is not a very strong function of V_{th} and thus appreciable variation in I_{on} is not observed. In this work, we aim to model this functional dependence by proposing a location-dependent V_{th} model which is able to model the current 'kick-back' effect shown in the figure.

Typically, fabrication facilities do not release TCAD setups, therefore the model must be such that it can be developed entirely using SPICE data. However, in this work, we wish to verify our results against TCAD simulation. Since matching TCAD and SPICE models are not easily available, we have performed our modeling based on TCAD data. However, this data is treated *exactly* as if the values were taken from SPICE results. In other words, we use only the I vs. W characteristics to perform the

fitting, which are easily obtained from SPICE simulations. TCAD data is used only for the purpose of providing a meaningful comparison point. We try to make this TCAD setup as close as possible to an industrial 90 nm technology as described later.

Referring back to the I_{off} vs. W plot, we note that after a width of 100 nm, the current is linear with width. This suggests that for points further than 50nm from each edge, the threshold voltage is constant. However, below 100 nm, the flattening of the curve indicates that this is the point where the effects from the two edges interact. We conclude that the edge effect extends to approximately 50nm on each side. This conclusion is confirmed by the Jon/Joff vs. location plot generated by TCAD simulation shown in Figure V.8. The threshold voltage is thus modeled as a piecewise function of location.

We now discuss the process of determining the functional form of the threshold voltage, f(x), in the edge affected region.

The surface potential, and hence the threshold voltage at a location is a complicated function of distance from the edge [93, 99]. The I_{off} density is known to be an exponential function of V_{th}. Since the only information available is the total current, which is the integral of the current density over the width, fitting is to be performed on this quantity. Since a function can always be approximated as a polynomial, we find the highest order polynomial whose exponential can be integrated to obtain standard functions. Mathematica [100] is able to integrate functions of the form *exp(K1*x2+K2*x)* conveniently. We also find that the quadratic model captures the 'kick-back' effect correctly. The functional form of the V_{th} is described in equations (V.1), (V.2) and (V.3). Where the edge effects overlap, the quadratics add up to create a stronger equivalent effect. Here, $V_{th}(x)$ is the threshold voltage as a function of location in the edge affected region. $V_{th}(middle)$ is the flat threshold voltage at distances beyond the edge region. The total width of the device is W, the width of the edge region is w and *K1, K2* are

empirically fitted constants. For $W >= 2w$

$$\begin{aligned} V_{th}(x) &= V_{th}(middle) - K1 \times (x-w)^2 \\ &+ K2 \times (x-w) \quad 0 <= x <= w \\ V_{th}(x) &= V_{th}(middle) \quad w <= x <= W-w \\ V_{th}(x) &= V_{th}(middle) - K1 \times (W-x-w)^2 + \\ &K2 \times (W-x-w) \quad W-w <= x <= W \end{aligned} \tag{V.1}$$

For $w <= W <= 2w$

$$\begin{aligned} V_{th}(x) &= V_{th}(middle) - K1 \times (x-w)^2 + K2 \times (x-w) \quad 0 <= x <= W-w \\ V_{th}(x) &= V_{th}(middle) - K1 \times (x-w)^2 + K2 \times (x-w) - K1 \times (W-x-w)^2 \\ &+ K2 \times (W-x-w) \quad W-w <= x <= w \\ V_{th}(x) &= V_{th}(middle) - K1 \times (W-x-w)^2 + \\ &K2 \times (W-x-w) \quad w <= x <= W \end{aligned} \tag{V.2}$$

For $W <= w$

$$V_{th}(x) = V_{th}(middle) - K1 \times (x-w)^2 + K2 \times (x-w) - K1 \times (W-x-w)^2 + K2 \times (W-x-w) \tag{V.3}$$

Figure V.7 show plots of V_{th} vs. location for a particular choice of *K1* and *K2* for $W> = 2w$ and $w< =W< = 2w$ respectively. The fitting is performed using Mathematica. The coefficients are adjusted to match the total current (integral of the density) to the observed values shown in Figures V.6, V.5. The procedure is repeated for different lengths and a new model is generated for each length from -10nm to +10nm around the nominal value.

When a non-rectilinear gate is encountered, it is split into several rectangles, and for each rectangle the model corresponding to its length is chosen. The limits of integration are dependent on the position of the rectangle along the width. Since the current varies along the width, integrating over different sections yields different values of total current, and the location dependence is captured.

The accuracy of the fitting model is shown for L=90 nm in Figure V.5 and Figure V.6. Although the fitting is not very accurate at low values of width (it is

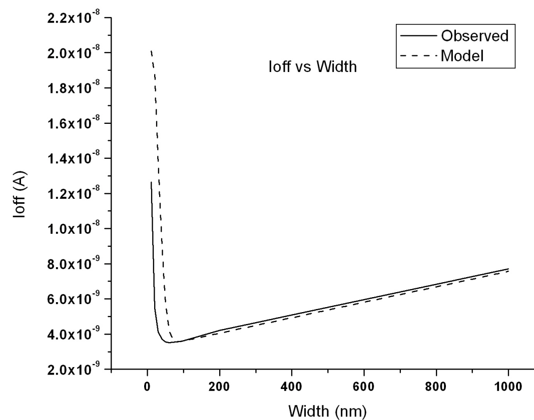


Figure V.5: Comparison of I_{off} vs. width characteristics of observed data and fitted model.

not possible to capture all the device parameters in this simplistic model), the trend is correctly predicted and the effective current overhead for non-rectilinear gates is still fairly accurate as discussed in the results section. It is also useful to note that the largest discrepancies are for very small values of width, which are well below the minimμm sizes allowed by the process. The model behaves similarly for different values of L.

Figure V.6 shows the behavior of the model for I_{on}. It should be noted that the V_{th} fit is entirely based on matching I_{off} values, as this parameter is considerably more sensitive to V_{th}. The I_{on} values are calculated based on the V_{th} values predicted by the model.

V.B.3 TCAD Setup for Model Accuracy Verification

To verify the accuracy of the models derived previously, a 3D TCAD simulation tool Synopsys Davinci [99] is used. We generate a single N-Channel MOSFET as a baseline device with L_nominal = 90 nm and Width = 400nm. To include fully-recessed oxide isolation (e.g., STI) in the simulation, we surround the devices with oxide material which is 0.1μm in depth and 0.1um in width. Initial parameters of the device are taken from the ITRS 90 nm technology node13. The values are then tuned to meet the device characteristics (i.e., I_{on} and I_{off}) for an industrial 90 nm technology obtained through

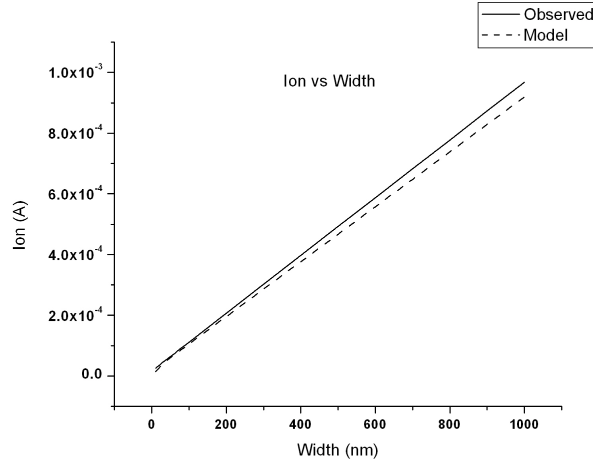


Figure V.6: Comparison of I_{on} vs. width characteristics of observed data and fitted model.

Table V.5: TCAD model parameters.

Parameters	**Value**
L_nominal	90 nm
Width	0.4μm
Vdd	1.2V
Tox	1.82nm
Channel doping	3.5e+18 cm^{-3}
NSUB	3.0e+15 cm^{-3}
Junction depth	0.03μm
Line-End Extension	0.1μm
S/D Electrode length	0.07μm
S/D Electrode width	0.4μm
S/D Region to Gate poly	0.02um
STI width	0.1μm
STI depth	0.1μm

SPICE simulation. The final parameters we use to measure I_{on} and I_{off} of non-rectilinear devices are shown in Table V.5. I_{off} and I_{on} comparisons of 3D TCAD simulation results with HSPICE are shown in Figure V.9. As can be seen from the plot, the discrepancy between Davinci simulation and HSPICE slightly increases as the device width increases.

To check the current dependency, thus V_{th} dependency on the location along the device width, we measure the current density along the width. Figure V.8 shows the

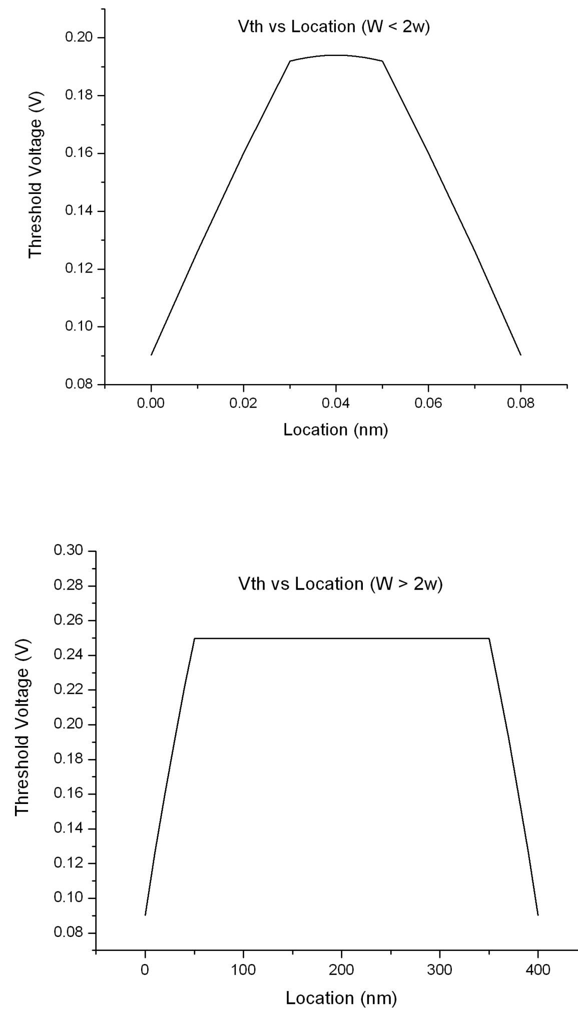


Figure V.7: V_{th} as a function of location.

current density value along the device width (i.e., z-axis in Figure V.8). As can be seen from the figure, current density along the device width represents a "tub-shaped" plot; the current through most of device is constant (i.e., flat along the z-axis) but it increases drastically over a distance approximately 50nm from device edge. These observations confirm the conclusions arrived at previously in this section of the chapter.

To simulate the device performance of non-rectilinear devices we introduce an irregularity and sweep the location, width, and length of the protrusion or depression. Figure V.10 shows a basic gate structure which we use for the 3D device simulation.

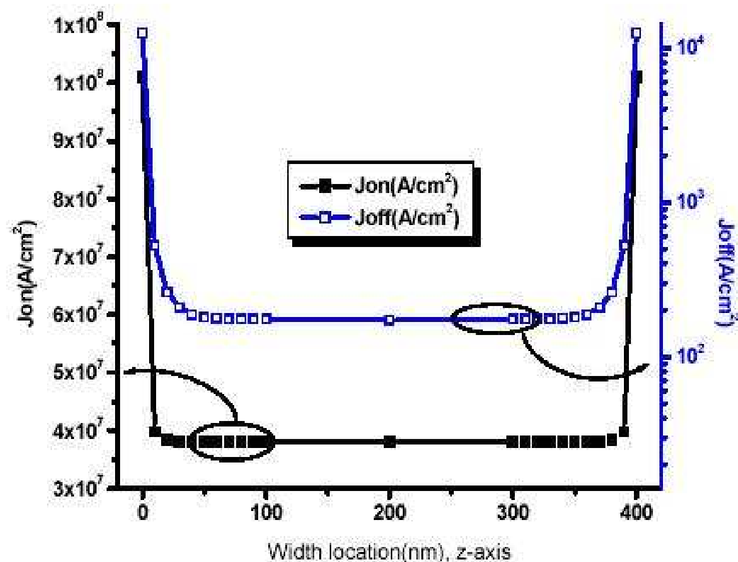


Figure V.8: On and Off current densities as a function of position along channel.

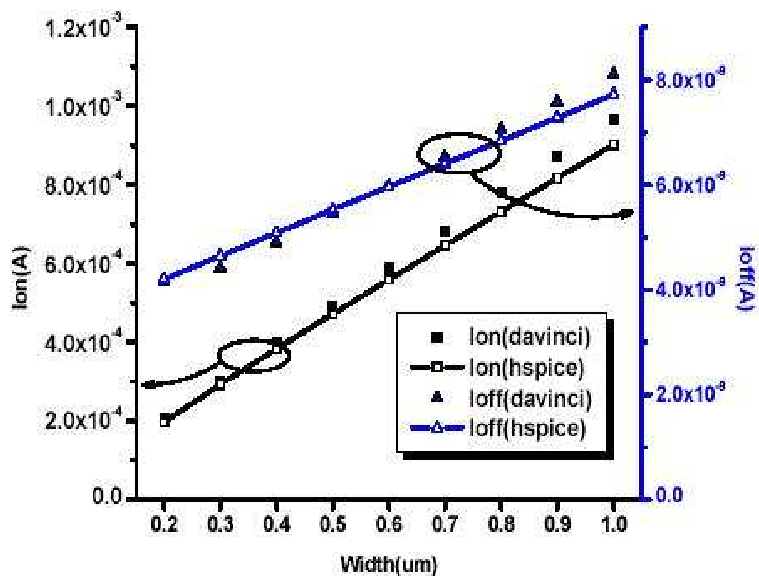


Figure V.9: I_{on} and I_{off} as a function of width for SPICE and Davinci setups.

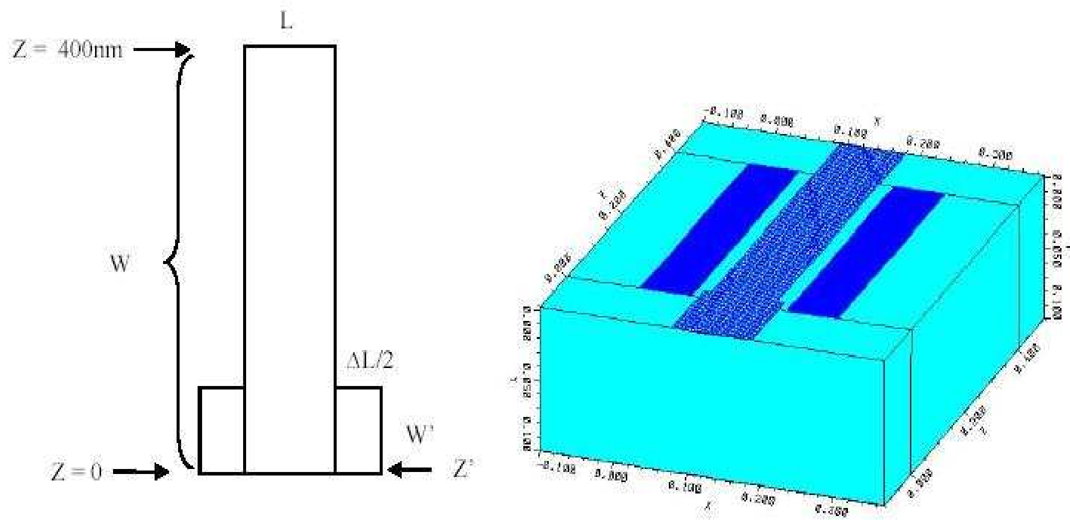


Figure V.10: Davinci structure used for verification of results diagrammatic representation and 3D Device Structure.

As can be seen from the figure, $\Delta L/2$ is the half protrusion length, W' is the width of protrusion, and Z' is defined as the bottom location of the protrusion. We sweep ΔL from -10nm to +10nm with a step of 2nm, W' from 20nm to 200nm with a step of 20nm, and Z' from 0 (i.e., at the bottom edge of device) to 100 nm (i.e., center of device).

Positive ΔL corresponds to a protrusion and negative ΔL corresponds to a depression. The line-end extension profile also follows the shape of the edges of devices (e.g., if there is a protrusion at the bottom edge, the length of the line-end extension is the same as the length of the protrusion). A contour plot of a 3D device mesh structure is shown in Figure V.10 where $\Delta L/2$ =+10nm, W'=20nm, and Z'=0.

V.B.4 Results

In this subsection, we describe the results of our comparison with TCAD simulations. The comparison structure is shown in Figure V.10. Table V.6 shows results for various values of $\Delta L/2$ and Z', for W = 400nm, W' = 20nm. Table V.7 shows the same results for W'=40nm. The accuracy is compared against a flat model where the threshold voltage is assumed to be the same across the entire device. The flat model is similar to that used in previous literature [89, 90, 91]. It is found that the variation in

Table V.6: Results for non-rectilinear gates with protrusion or depression of width 20nm.

Width of Protrusion/Depression = 20nm						
	Length of Protrusion	82	86	90	94	98
Location(nm)						
0	Davinci	1.477	1.163	1.000	0.919	0.870
	Proposed Model	***1.543***	***1.194***	***1.000***	***0.911***	***0.838***
	Flat Model	1.133	1.040	1.000	0.978	0.967
20	Davinci	1.247	1.083	1.000	0.982	0.971
	Proposed Model	***1.197***	***1.061***	***1.000***	***0.970***	***0.954***
	Flat Model	1.133	1.040	1.000	0.978	0.967
40	Davinci	1.124	1.042	1.000	0.991	0.984
	Proposed Model	***1.088***	***1.024***	***1.000***	***0.989***	***0.984***
	Flat Model	1.133	1.040	1.000	0.978	0.967
60	Davinci	1.092	1.031	1.000	0.993	0.989
	Proposed Model	***1.079***	***1.021***	***1.000***	***0.991***	***0.986***
	Flat Model	1.133	1.040	1.000	0.978	0.967
80	Davinci	1.082	1.027	1.000	0.994	0.991
	Proposed Model	***1.079***	***1.021***	***1.000***	***0.991***	***0.986***
	Flat Model	1.133	1.040	1.000	0.978	0.967

Table V.7: Results for non-rectilinear gates with protrusion or depression of width 40nm.

Width of Protrusion/Depression = 40nm						
	Length of Protrusion	82	86	90	94	98
Location(nm)?						
0	Davinci	1.660	1.217	1.000	0.880	0.800
	Proposed Model	***1.741***	***1.255***	***1.000***	***0.872***	***0.792***
	Flat Model	1.267	1.080	1.000	0.957	0.935
20	Davinci	1.340	1.111	1.000	0.961	0.930
	Proposed Model	***1.286***	***1.085***	***1.000***	***0.959***	***0.938***
	Flat Model	1.267	1.080	1.000	0.957	0.935
40	Davinci	1.193	1.064	1.000	0.976	0.958
	Proposed Model	***1.168***	***1.045***	***1.000***	***0.980***	***0.970***
	Flat Model	1.267	1.080	1.000	0.957	0.935
60	Davinci	1.156	1.051	1.000	0.981	0.968
	Proposed Model	***1.158***	***1.042***	***1.000***	***0.981***	***0.972***
	Flat Model	1.267	1.080	1.000	0.957	0.935
80	Davinci	1.145	1.046	1.000	0.983	0.971
	Proposed Model	***1.158***	***1.042***	***1.000***	***0.981***	***0.972***
	Flat Model	1.267	1.080	1.000	0.957	0.935

I_{off} is considerable based on the location of the length variation and the flat model completely fails to predict this. The proposed model, on the other hand, is able to capture this dependency very well.

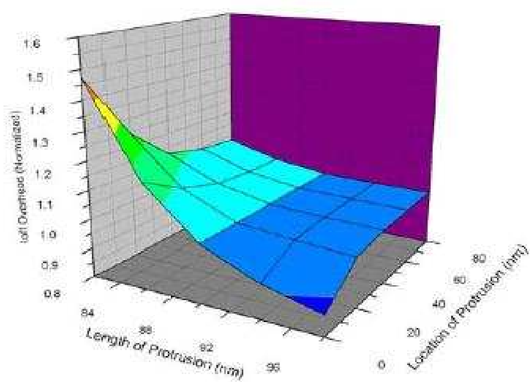
The data in the tables is shown in Figure V.11 in graphical form as contour plots to indicate the dependence of I_{off} on the location and the length of the irregularity. Plots for the model as well as the actual data are shown. Examining the plots reveals that the overhead trend is closely captured by our model. On the other hand, the flat model is completely unable to capture the large dependence on location.

Since the dependence of I_{on} on threshold voltage is very weak, we find no appreciable difference in I_{on} overheads for irregularities at different locations. Therefore, this model does not help particularly for I_{on} and we do not show results for the same.

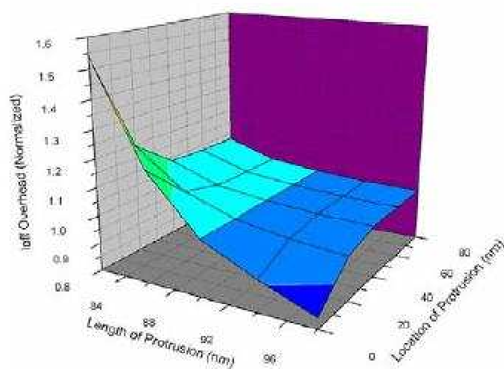
V.B.5 Conclusions

To the best of our knowledge, this work proposes the first location-dependent threshold voltage model to analyze irregular transistor gates due to imperfect lithographic patterns. These gates are found to have distinctly different I_{off} and I_{on} characteristics compared to perfectly rectangular gates, and cannot be handled correctly by SPICE-like circuit analysis tools. We have recognized the fact that the threshold voltage is non-uniform across the device width and incorporated this effect into our model. It is found that the location of a particular irregularity along the channel significantly affects its impact on device characteristics. We have proposed a threshold voltage model that is simultaneously dependent on the location of a point along the channel as well as the length at that point. Using this

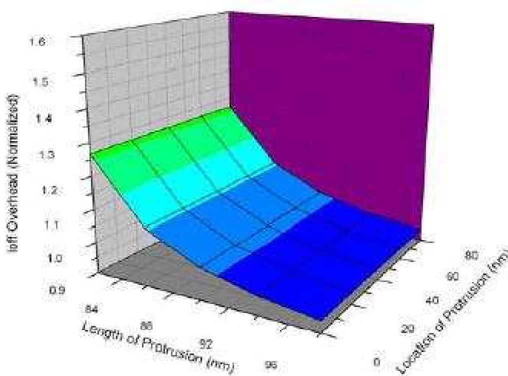
model, we analyze non-rectilinear gates with various structures. Comparing our results with TCAD simulation results, we show that this method is considerably more accurate than a method that considers the threshold voltage to be flat along the channel. Future work in this area concentrates on improving the accuracy of this model and obtaining matching SPICE and TCAD setups for pure SPICE- based modeling.



(a) Contour plot of observed data



(b) Contour plot of fitted model



(c) Contour plot of flat model

Figure V.11: I_{off} contours vs. protrusion dimensions.

V.C Acknowledgments

Chapter V is in part a reprint of "Lithography Simulation-Based Full-Chip Design Analyses", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2006 and "Modeling of Non-Uniform Device Geometries for Post-Lithography Circuit Analysis", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2006. I would like to thank my coauthors Puneet Sharma, Saumil Shah, Youngmin Kim, Dr. O. Sam Nakagawa, Dr. Dennis Sylvester and Dr. Andrew B. Kahng.

VI

The Cost Angle

Continued technology scaling in the subwavelength lithography regime results in printed features that are substantially smaller than the optical wavelength used to pattern them. For instance, modern 130 nm CMOS processes use 248 nm exposure tools, and the industry roadmap through the 45 nm technology node will use 193 nm (immersion) lithography. The International Technology Roadmap for Semiconductors (ITRS) [118] identifies aggressive microprocessor (MPU) gate-lengths and highly controllable gate CD control as two critical issues for the continuation of Moore's Law cost and integration trajectories. To meet ITRS requirements (see Table VI.1), resolution enhancement techniques (RETs) such as optical proximity correction (OPC) and phase-shift masks (PSM) are applied to an increasing number of mask layers and with increasing aggressiveness. The recent steep increase in mask costs and lithographic complexity due to these RET approaches has had a harmful impact on design starts and project risk across the semiconductor industry. Cost of ownership (COO) has become a key consideration in adoption of various lithography technologies.

VI.A Performance-Driven OPC for Mask Cost Reduction

The increasing application of RETs makes mask data preparation (MDP) a serious bottleneck for the semiconductor industry: figure counts explode as dimensions shrink and RETs are used more heavily. Compared with the mask set cost in 0.35 μm, the cost at the 0.13 μm generation with extensive PSM implemented is four times

larger [153]. Figure counts, corresponding to polygons as seen in the IC layout editor grow tremendously due to sub-resolution assist features and other proximity corrections. Increases in the fractured lay-out data volume lead to disproportionate increases in mask writing and inspection time. According to the 2005 ITRS [118], the maximum single-layer MEBES file size increases from 64GB in 130 nm to 216GB in 90 nm. Another observation concerns the relationship between design type and lithography costs, namely, that the total cost to produce low-volume parts is dominated by mask costs [148]. Half of all masks produced are used on less than 570 wafers (this translates roughly to production volumes of $\leq$ 100,000 parts). At such low usages, the high added costs of RETs cannot be completely amortized and the corresponding cost per die becomes very large. Thus, designers and manufacturers are jointly faced with determining how best to apply RETs to standard-cell libraries to minimize mask cost.

In this work we focus on OPC, which is a major contributor to mask costs as well as design turnaround time (TAT). More than a 5X increase in data volume and several days of CPU runtime are common side effects of OPC insertion in current designs [154]. With respect to the cost breakdown shown in Figure VI.1, OPC affects mask data preparation (MDP), defect inspection (and implicitly defect repair), and the mask-writing process itself. Today, variable-shaped electron beam mask writers, in combination with vector scanning[1], comprise the dominant approach to high-speed mask writing. In the standard mask data preparation flow, the input GDSII layout data is converted into the mask writer format by *fracturing* into rectangles or trapezoids of different dimensions. With OPC applied during mask data preparation, the number of line edges increases by 4-8X over a non-OPC layout, driving up the resulting GDSII file size as well as fractured data (e.g., MEBES format) volume [155]. Mask writers are hence slowed by the software for e-beam data fracturing and transfer, as well as by the extremely large file sizes involved. Moreover, increases in the fractured layout data volume[2] lead to disproportionate, super linear increases in mask writing and inspection

[1]Compared to traditional raster scanning, vector scanning allows features to be scaled up or down in size while maintaining sharpness, but the write cost is proportional to feature complexity: the mask pattern must be decomposed into a set of disjoint “shots” or “flashes”, each of which takes roughly constant (unit) time.

[2]E.g., according to the 2005 ITRS [118], the maximum single-layer MEBES file size increases from 216GB in 90 nm to 729 GB in 65 nm.

Table VI.1: The ITRS requirement of gate dimension variation control is becoming more stringent as the technology scales.

Year	2005	2007	2010	2013
Technology Node	90 nm	65 nm	45 nm	32 nm
MPU gate-length	32 nm	25 nm	18 nm	13 nm
MPU gate-length 3σ	3.3 nm	2.6 nm	1.9 nm	1.3 nm

time. Compounding these woes is the fact that the total cost to produce low-volume parts is now dominated by mask costs [148] since masks costs cannot be amortized over a large number of shipped products. There is a clear need to reduce the negative implications of OPC on total design cost while maintaining the printability improvements provided by this crucial RET step.

Design Function in the Design-Manufacturing Interface A primary failing of current approaches to the design-manufacturing interface is in lack of communication across disciplines and/or tool sets. For example, it is well documented that mask writers do not differentiate among shapes being patterned - given this, gates in critical paths are given the same priority as pieces of a company logo and errors in either of these shapes will cause mask inspection tools to reject a mask. In this light, we observe that OPC has traditionally been treated as a purely geometric exercise wherein the OPC insertion tool tries to match every edge as best as it can. As we show in our work, such "overcorrection" leads to higher mask costs and larger runtimes.

A Performance-Driven OPC Methodology In this work, we propose a performance-driven OPC methodology that is demonstrated to be highly implementable within the limitations of current industrial design flows. Contributions of our work include the following.

- *Quantified CD error tolerance.* We propose a mathematical programming based budgeting algorithm that outputs edge placement error tolerances (in nm) for layout features.

- *Integration within a commercial MDP flow.* We describe a practical flow implementable with commercial tools and validate the minimum cost of correction methodology.

Table VI.2: Correspondence between the traditional gate sizing problem and the minimum cost of correction (to achieve a prescribed selling point delay with given yield) problem.

Gate Sizing		MinCorr
Area	$\equiv$	Cost of Correction
Nominal delay	$\equiv$	Delay $\mu + k\sigma$
Cycle time	$\equiv$	Selling point delay
Die area	$\equiv$	Total Cost of OPC

- *Reduction of OPC overhead.* We measure OPC overhead in terms of additional MEBES features as well as runtime of the OPC insertion tool and show substantial improvements in both.

VI.A.1 General Cost of Correction Flow (MinCorr) Based on Sizing

We describe a generic yield closure flow which is very similar to traditional flows for timing closure. In this subsection, we describe the elements of such a flow.

In this generic *sizing based MinCorr* flow, we emphasize the striking similarity to conventional timing optimization flows. The key analogy - and assumption - is that there are discrete allowed "sizes" in the MinCorr problem that correspond to allowed levels of OPC aggressiveness (see Figure VI.3). Furthermore, for each instance in the design there is a cost and delay penalty associated with every level of correction. The mapping between traditional gate-sizing and the MinCorr problems is reproduced in Table VI.2. This flow involves construction of cost/yield aware libraries for each level of correction, and a commercial STA tool together with a selling point yield bonding algorithm which applies timing driven cost optimization. We acknowledge the following facts during the flow development process:

- We assume that different levels of OPC can be independently applied to any gate in the design. Corresponding to each level of correction, there is an effective channel length L_{eff} variation and an associated cost.

- Differentiate field-poly from gate-poly features. Field poly features do not impact performance and hence any delay-constrained MinCorr approach should not change

the correction of field-poly. Moreover, quality metrics of field-poly are different from those of gate-poly (e.g., contact coverage). By recognizing these two types of poly features, we may avoid over estimating cost savings achieved with this approach.

- The figure count numbers (which are proxies for mask cost implications) for the cells are extracted from post-OPC layout with commercial OPC insertion tool.

- OPC corrects the layout for pattern-dependent through-pitch CD variation. Such variations are predictable, for example, by lithography simulations.

With these facts, the *MinCorr* problem is summarized as: given a range of allowable corrections for each feature in the layout as well as the cost and CD deviation associated with each level of correction, find the level of correction for each feature such that prescribed circuit performance is attained with minimum total correction cost. Commercial OPC tools are driven by *edge placement errors* (EPEs), rather than critical dimensions (CDs). Thus, we specify a practical *MinCorr* with a practical implementation - *EPEMinCorr*. We can summarize the key contribution of EPEMinCorr as: *we devise a flow to pass design constraints on to the OPC insertion tool in a form that it can understand.*

As previously mentioned, OPC insertion tools are driven by *edge placement error* (EPE) *tolerances* (e.g., Figure VI.3 shows OPC layers driven by different EPE requirements). Typical model-based OPC techniques break up edges into *edge-fragments* that are then iteratively shifted outward or inward (with respect to the feature boundary) based on simulation results, until the estimated wafer image of each edge-fragment falls within the specified EPE tolerance. EPE (and hence EPE tolerance) is typically signed, with negative EPE corresponding to a decrease in CD (i.e., moving the edge inward with respect to the feature boundary). An example of a layout fragment and its EPE is shown in Figure VI.2. Mask data volume is heavily dependent on the assigned EPE tolerance that the OPC insertion tool is asked to achieve. For example, Figure VI.4 shows the change in MEBES file size for cell with applied OPC as the EPE tolerance is varied. In this particular example, loosened EPE tolerances can reduce data volume by roughly 20% relative to tight control levels.

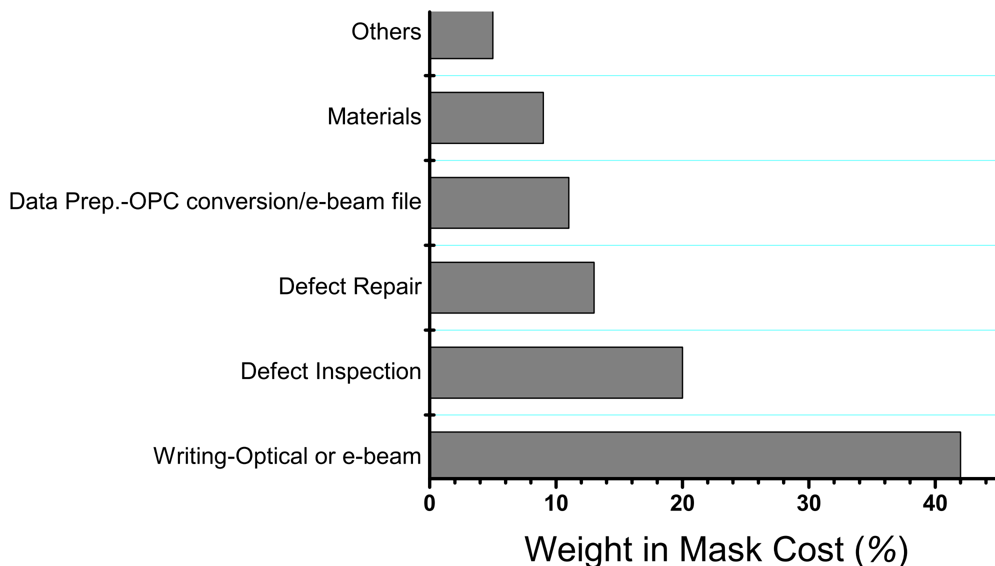
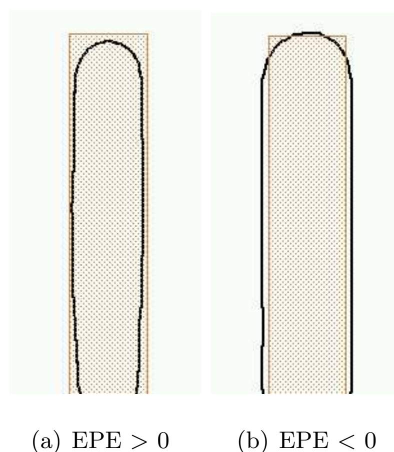


Figure VI.1: Relative contributions of various components of mask cost [157].



(a) EPE > 0 (b) EPE < 0

Figure VI.2: The signed edge placement error (EPE).

Since model-based OPC corrects for pattern-dependent CD variation, which is systematic and predictable, we assert that OPC actually determines *nominal timing*. This allows us to base our OPC insertion methodology on traditional corner-case timing analysis tools instead of (currently non-existent from a commercial standpoint) statistical timing analysis tools. Our methodology adopts a slack budgeting based approach - as opposed to the sizing based approach as mentioned above - to determine EPE tolerance values for every feature in the design. For simplicity, our description and experiments reported here are restricted in two ways: (1) we apply selective EPE tolerances in OPC

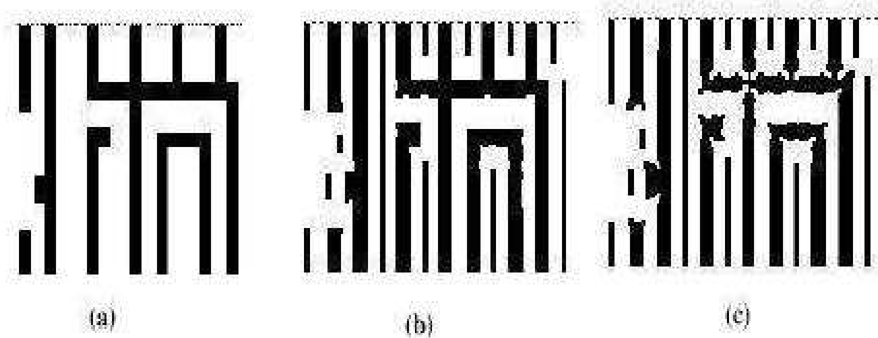


Figure VI.3: An example of three levels of OPC [158]. (a) No OPC, (b) Medium OPC, (c) Aggressive OPC.

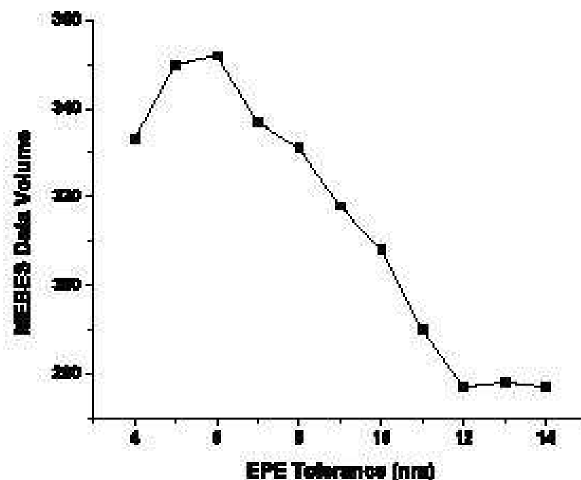


Figure VI.4: Mask data volume (kB) vs. EPE tolerance for a NAND3X4 cell in TSMC 130 nm technology.

to only gate-poly features, and (2) every gate feature in a given cell instance is assumed to have the same EPE tolerance (the approach may be made more fine-grained using the same techniques that we describe). Figure VI.5 shows our EPEMinCorr flow. The quality of results generated by the flow are measured as MEBES data volume of fractured post-OPC insertion layout shapes as well as OPC insertion tool runtime, which can be prohibitive when run at the full-chip level. We now describe details of the major steps

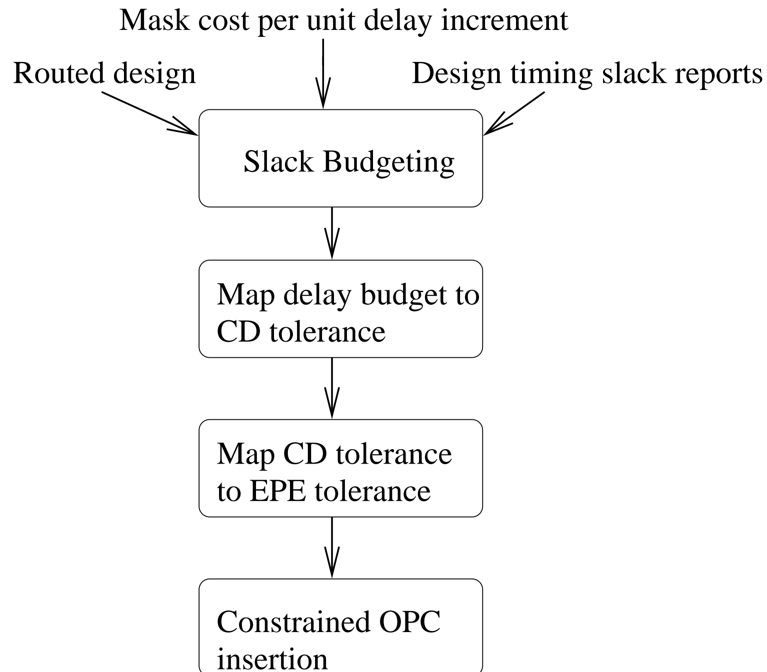


Figure VI.5: The EPEMinCorr flow to find quantified edge placement error tolerances for layout features and drive OPC with them.

of the Figure VI.5 EPEMinCorr flow.

Slack Budgeting

The slack budgeting problem seeks to distribute slack at the primary inputs of combinational logic (i.e., sequential cell outputs) to various nodes in the design. One of the earliest and simplest approaches, the zero-slack algorithm (ZSA) [159], iteratively finds the minimum-slack timing path and distributes its slack equally among the nodes in the path. The MISA algorithm for slack budgeting proposed in [160] distributes slack iteratively to an independent set of nodes. As with ZSA, the objective is to maximize the total added incremental delay budget on timing arcs. A weighted version of MISA is also proposed in [160].

We observe:

- Neither MISA variant is guaranteed to provide optimal solutions.
- ZSA is much faster than MISA, and a weighted version of ZSA can also be formu-

lated.

- While [161] formulates the budgeting problem as a convex programming problem, full-chip MISA or mathematical programming is, as far as we can determine, too CPU-intensive for inclusion in a practical flow.

We propose to approximate full-chip mathematical programming by iteratively solving a sequence of linear programs (LPs). In each iteration, slack is budgeted among the top k available paths. Once a budget is obtained for a node, this budget is retained as an upper bound for subsequent iterations. The process is repeated until all nodes have been assigned a slack budget or path slack is sufficiently large. The basic LP has the following form:

$$\text{Maximize } \sum_{i=1}^{n} C_i s_i \tag{VI.1}$$
$$\sum_{j \in P_k} s_j \leq S_k \ \forall \ k \in \text{Current path list}$$
$$s_j \leq s_j^f \ \forall \ j \in F$$

where C_i denotes the correction cost decrease per unit delay increase for cell i, and s_i is the slack allocated to cell i. The notation P_k is used to denote the kth most critical path, and S_k is the slack of this path. Finally, F denotes the set of nodes with slacks fixed from previous iterations. An example sequence of LPs might be obtained by allowing k to take on the range from 1 to 100 in the first iteration, 101 to 200 in the second iteration, and so on.

We observe that when a budgeting formulation is adopted in place of a sizing formulation, the method of accounting for changes in next-stage input pin capacitance becomes an open question. To be conservative, we generate timing reports with pin input capacitances that correspond to the loosest tolerance (i.e., largest pin capacitance) but gate delays corresponding to the tightest achievable tolerance. C_i is obtained via a pre-built look-up table (similar to .lib format) containing the increase in data volume, mapped against delay change.

Our budgeting procedure yields positive delay budgets leading to positive EPE tolerances. Since EPE tolerance is a signed quantity (e.g., in Mentor Calibre, a common

OPC insertion tool), negative EPE tolerances (corresponding to reduced gate-length and faster delay) can also be obtained in a similar way based on hold-time or leakage power constraints. However, in this work we assume equal positive and negative EPE tolerances since we deal with purely combinational benchmarks and focus on timing rather than power.

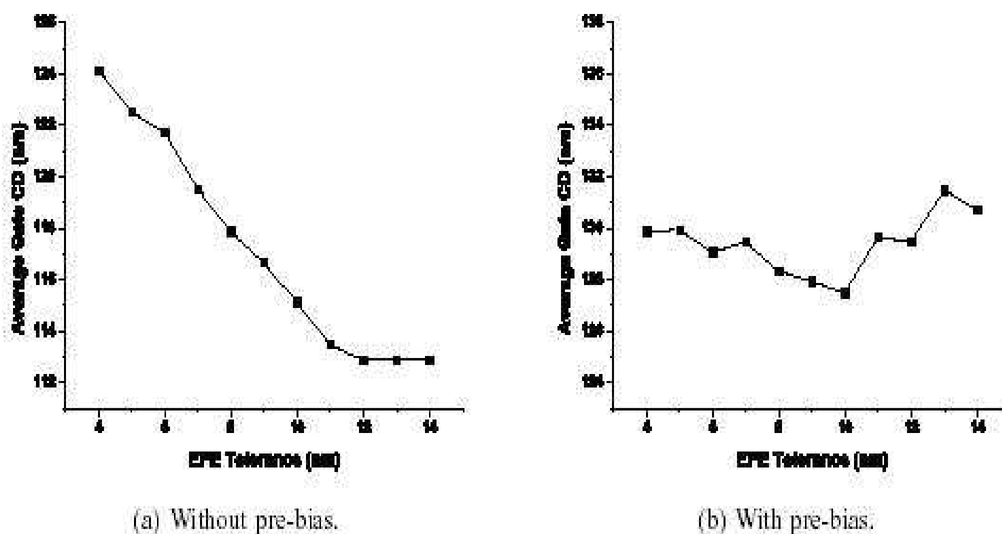
Calculation of CD Tolerances

To map delay budgets found from the above linear programming based formulation to CD tolerances, we require characterization of a standard-cell library with varying gate-lengths. Using such an augmented library, along with input slew and load capacitance values for every cell instance, we can map delay budgets to the corresponding gate-lengths. For example, if a particular instance with specified load and input slew rate has a delay budget of 100ps, then we can select the longest gate-length implementation of this gate type that meets this delay. This largest allowable CD will lead to a more easily manufactured gate with less RET effort. Subtracting these budgeted gate-lengths from nominal gate-lengths yields the CD tolerance for every cell in the design.

Calculation of EPE Tolerances

The next step in our flow maps CD tolerances to signed EPE tolerances. Again, obtaining EPE tolerances is crucial since this is the parameter which OPC insertion tools understand and can exploit. As noted above, in this work we assume positive and negative EPE tolerance to be the same. Since CD is determined by two edges, the worst-case CD tolerance is twice the EPE tolerance.

In most lithography processes, gates shrink along their entire width such that the printed gate-length is always smaller than the drawn gate-length, except at the corners of the critical gate feature. OPC typically biases the gate length such that corrected gate-length is *larger* than the designer-drawn gate-length. Thus, model-based OPC shifts edges *outward*, i.e., in the "positive" direction, until it meets the EPE tolerance specification. If the step size of each edge move is small enough, the EPE along the gate width will always be negative (since we are approaching the larger nominal gate-length value starting from the smaller printed gate-length value). As a result, actual printed



(a) Without pre-bias. (b) With pre-bias.

Figure VI.6: Comparison of average printed gate CD with and without pre-bias for the cell macro NAND3X4.

gate-length will almost always be smaller than the drawn gate-length, leading to leakier but faster devices.

To achieve a more unbiased deviation from nominal, we exploit the behavior of the OPC tool by applying simple pre-biasing of gate features in an attempt to achieve EPE tolerances that are equal to CD tolerance. Specifically, we pre-bias each gate feature by its intended EPE tolerance. For instance, for a drawn gate-length of 130 nm and EPE tolerance of 10 nm, the printed CD would typically lie between 110 nm and 130 nm (each edge shifts by 10 nm inward). If the gate-length is biased by 10 nm so that the OPC tool views 140 nm as the target CD, the printed CD would lie between 120 nm and 140 nm, which amounts to a ± 10 nm CD tolerance. In this way, pre-biasing achieves CD tolerances equal to the EPE tolerance. An example of the average CD for a specific gate-poly with and without pre-biasing is shown in Figure VI.6. It is clear that pre-biasing achieves its goal of attaining average CDs that are very close to the target CD (130 nm in our case). Another point illustrated in Figure VI.6 is that the variation in CD (measured as the standard deviation of CD taken across all edge-fragments) grows as the EPE tolerance is relaxed. This is shown more clearly in Section VI.A.2.

Table VI.3: Benchmark details.

Test Case	Source	Cell Count
c432	ISCAS85	337
c5315	ISCAS85	2093
c6288	ISCAS85	4523
c7552	ISCAS85	2775
alu128	Opencores	12403
r4_sova	Industry	34288

Constrained OPC

We enforce the obtained EPE tolerances within a commercial OPC insertion flow. We use *Calibre* [162] as the OPC insertion tool; details of constraining the tool are described in the next subsection.

VI.A.2 Experimental Setup and Results

In this section we describe our experiments and the results obtained in order to validate the EPEMinCorr methodology.

Test Cases

We use several combinational benchmarks drawn from ISCAS85 suite of benchmarks and Opencores [163]. These benchmark circuits are synthesized, placed and routed in a restricted TSMC 0.13 μm library containing a total of 32 cell macros with cell types of BUF, INV, NAND2, NAND3, NAND4, NOR2, NOR3, and NOR4. The test case characteristics are given in Table VI.3.

Library Characterization

We assume a total of EPE tolerance levels ranging from ±4 nm to ±14 nm. Corresponding to each EPE tolerance, the worst case gate-length is $130nm{+}EPE_Tolerance$. We map cell delays to EPE tolerance levels by creating multiple .lib files for each of the 10 worst case gate-lengths using circuit simulation. For simplicity, we neglect the dependence of delay on input slew in our analysis but this could easily be added to the framework.

Expected mask cost for each cell type is extracted as a function of EPE tolerance. We run model-based OPC using Calibre on individual cells followed by fracturing to obtain MEBES data volume numbers for each (cell, tolerance) pair. Though the exact corrections applied to a cell will depend somewhat on its placement environment, standalone OPC is fairly representative of data volume changes with changing EPE tolerance. Finally, we calculate the sensitivity of mask cost to delay change under the assumption that cost reduction is a linear function of delay increase. This assumption is based on linearity between gate delay and CD as well as the rough linearity shown in Figure VI.4 between data volume and EPE tolerance. We then build a .lib-like look-up table of correction cost sensitivities (with respect to the tightest EPE tolerance of 4 nm). When slack is distributed to various nodes, we extract the load capacitances that are used to identify entries in the sensitivity table. Cost change is most sensitive to delay changes when the load capacitance is small (this typically indicates a small driver and subsequently small amount of data volume) and the sensitivity numbers are on the order of 1X to 10X MEBES features per ps delay change.

EPEMinCorr with Calibre

Our OPC flow involves assist feature insertion followed by model-based OPC. The EPE tolerance is assigned to each gate by the *tagging* command within Calibre. As indicated in Figure VI.7, we first separate the entire poly layer into gate-poly and field-poly components. The field-poly tolerance is taken to be ±14 nm while gate-poly tolerance ranges from ±4 nm to ±14 nm. We tag the assigned EPE tolerance to cell names. In this way, we can track the EPE tolerance of each gate individually. We take 1 nm as our step size[3] when applying OPC to obtain very precise correction levels. We set the iteration number to the minimum value beyond which adding mask cost and CD distribution show little sensitivity to OPCs, which is found experimentally. After model-based OPC is applied, we perform "printimage" simulations in Calibre to obtain the expected as-printed wafer image of the layout. Average gate CD and its standard deviation are extracted from this wafer image. The corrected GDSII is fractured into

[3]Step size is the minimum perturbation to an edge that model-based OPC can make. Smaller step sizes lead to better correction accuracy at the cost of runtime.

MEBES using CalibreMDP. The total mask data volume is then determined based on the MEBES file sizes.

Results

We synthesize the benchmark circuits using *Synopsys Design Compiler*. Place and route is performed using *Cadence Silicon Ensemble*. *Synopsys Primetime* is used to output the slack report of the top 500 critical paths (not true for the biggest benchmark r4_sova where more paths are needed as discussed below) as well as the load capacitance for each driving pin. As noted above, STA is run with a modified 134 nm (tightest EPE tolerance) library with pin capacitances corresponding to 144 nm (loosest EPE tolerance) to remain conservative after slack budgeting. We use *CPLEX v8.1* [164] as the mathematical programming solver to solve the budgeting linear program. Two types of benchmarks are involved in our experiments: (i) large designs with a "wall" of critical paths, e.g., r4_sova in Table VI.3; and (ii) circuits with fairly small sizes, e.g., benchmarks except r4_sova. For (ii), a single iteration is efficient to solve the budgeting problem; for (i) however, more iterations may be necessary because some paths which are potentially critical but are not reported due to the constraint of maximum number of critical paths may become top critical later on as they are not treated as optimization objects by the slack budgeting algorithm, resulting in performance degradation. One possible solution to this problem is to perform iterations to selectively include those paths that may cause performance degradation, as slack budgeting objects. Another simple but not as efficient option is to increase the constraint of maximum number of critical paths in the slack report. We deploy a hybrid way for r4_sova in our case, i.e., the constraint on the initial number of critical paths is increased from 500 to 10000, then in each iteration 5000 more paths that are potentially critical are included for slack budgeting. After 8 iterations the performance degradation due to the selective OPC is reduced to less than 1% (first iteration gives 4.3% performance degradation).

The extracted CD variation for test case *c*432 after EPEMinCorr OPC is shown in Figure VI.8. The distributions show that Calibre is able to enforce assigned tolerances very consistently. A tighter CD distribution for critical gates is achieved while non-critical gates (which can tolerate a larger deviation from nominal) have a more

Table VI.4: Impact of EPEMinCorr optimization on cost and CD. All runtimes are based on a 2.4GHz Xeon machine with 2GB memory running Linux.

Testcase	Traditional OPC Flow				EPEMinCorr Flow							
	CD Distribution All Gates (nm)		OPC Runtime (hr)	Delay (ns)	Budgeting Runtime (s)	CD Distribution				OPC Runtime (hr)	Delay (ns)	Normalized MEBES Volume
						All Gates (nm)		Critical Gates (nm)				
	mean	σ				mean	σ	mean	σ			
c432	126.8	1.57	0.2833	1.33	1	131.3	3.90	129.9	1.67	0.2047	1.33	0.83
c5315	126.1	1.82	1.261	1.94	3	131.7	4.70	129.7	1.89	1.180	1.94	0.79
c6288	126.0	1.37	3.564	5.21	9	131.4	4.45	129.7	1.27	2.697	5.21	0.82
c7552	126.2	1.89	1.986	1.59	4	132.0	4.77	130.1	1.99	1.428	1.59	0.78
alu128	126.1	1.48	14.31	3.28	11	131.5	4.93	130.8	2.04	9.215	3.28	0.76
r4_sova	126.3	2.07	39.72	8.19	29648	131.9	5.00	130.0	1.75	23.32	8.26	0.79

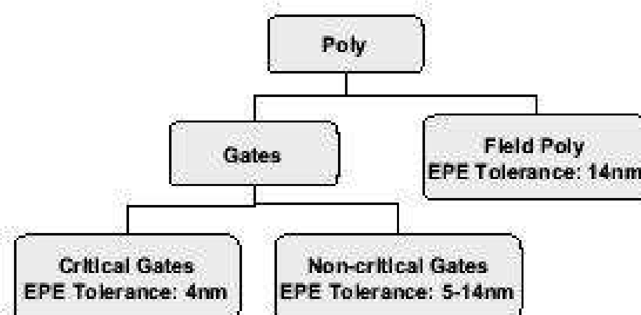


Figure VI.7: Summary of EPE assigment for OPC level control.

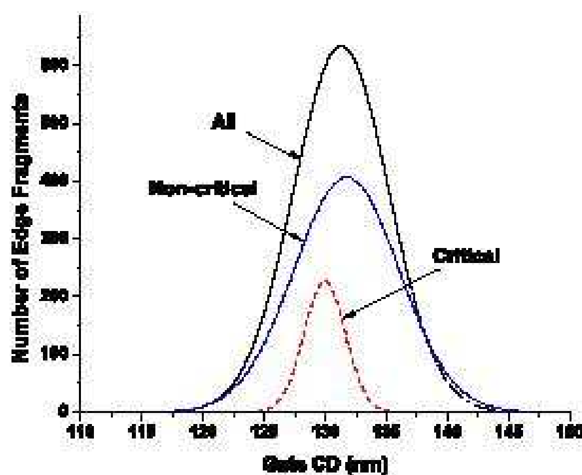


Figure VI.8: Gate CD distribution for c432. Gates with budgeted 4 nm EPE tolerance are labeled critical gates while others are labeled as non-critical. The y-axis shows the number of fragments of gate edges with a given printed CD.

relaxed (and hence less expensive to implement) gate-length distribution. Table VI.4 compares the runtime and data volume results for EPEMinCorr OPC and traditional OPC. For relatively small circuits, a single iteration of the budgeting approach ensures that there is no timing degradation going from the traditional to the EPEMinCorr flow, and the budgeting runtimes are negligibly small, ranging from 1s to 11s. For large designs especially those with a "wall" of critical paths, iterations may be required to avoid performance degradation and the sum of budgeting runtimes of each iteration may

reach several hours (7 hours for r4_sova). The important result is the amount of mask cost reductions achieved whether measures as runtime of model-based OPC or fractured MEBES data volume. EPEMinCorr flow reduces MEBES data volume by 17%-24%. Such reductions directly translate to substantial mask-write time improvements. OPC runtimes are improved by 6%-41%. These percentage numbers translate to a huge absolute TAT savings. For instance, the EPEMinCorr flow saves 16.4 hours compared to the traditional OPC flow on a 34000 gate benchmark.

VI.A.3 Conclusions and Future Work

We have proposed and implemented a practical means of reducing mask cost and the computational complexity of OPC insertion through formalized performance-driven OPC assignment. In particular we focus on the use of edge placement errors to drive OPC insertion tools and leverage EPEs as the mechanism to direct these tools to correct only to the levels required to meet timing specifications. An iterative linear programming based approach is used to perform slack budgeting in an efficient manner. This formulation results in a specific slack budget for each gate which is then mapped to allowable critical dimensions in the standard cell. Finally EPEs are generated from the CD budget and tags are placed on gates to indicate to the OPC insertion tool the appropriate level of correction. Our results on several benchmarks ranging from 300 to 34000 cells show up to 24% reductions in MEBES data volume which is frequently used as the metric for RET complexity. Furthermore, the runtime of the OPC insertion tool is reduced by up to 41% - this is critical since running OPC tools at the full-chip level is an extremely time-consuming step during the physical verification stage of IC design.

In future technologies allowable CD tolerances may be set more by bounds on acceptable leakage power than by traditional delay uncertainty constraints. We plan to incorporate power constraints into our formulation. Moreover, we plan to extend the EPEMinCorr methodology for field-poly features. Impact of field polysilicon shapes on performance comes from their overlap with contact layer. So field-poly extensions to EPEMinCorr will have to evaluate error in terms of contact coverage area. Expensive masking layers include diffusion, contact, metal1 and metal2 besides polysilicon. The performance impact of OPC errors on these other layers can also be computed and

consequently EPEMinCorr methodology extended.

Another direction of work is exploring other degrees of freedom in OPC besides EPE tolerance which have a strong effect on mask cost. Two such parameters are fragmentation and minimum jog length.

In a follow-up work of an industrial scale of application [165], a methodology similar to EPEMinCorr was used to optimize mask cost for a big design block. The resulting OPC'd layout went through dummy mask write at a mask shop. The authors reported 25% shot count reduction and up to 32% reduction in mask write time.

VI.B Modeling OPC Complexity for Design for Manufacturability

RETs such as assist feature insertion and OPC are mandatory post-tapeout steps to ensure printability of features in sub-90 nm technology nodes. Doubling of layout data volume every technology node combined with aggressive RET is driving mask set cost to prohibitive levels. Transferring design intent to OPC process can reduce the increasing complexity of masks in sub-90 nm technology nodes. Design intent-aware OPC applies different levels of OPC correction to different regions of a design based on their criticality.

There are two approaches to minimize mask cost using design information. In the first approach, timing and power analysis are performed on the design to identify all critical paths and their corresponding layout features. OPC is then performed with tight tolerances on all critical features and with relaxed tolerances on all non-critical features to minimize mask cost. This approach (e.g., Cote et.al [145]) does not modify the design flow prior to the tapeout. However, relaxing OPC tolerance uniformly on all non-critical features does not lead to the best possible mask cost reductions. In the second approach, tolerance optimization is performed by choosing OPC tolerance combination specific to the standard cell or wire pattern by analyzing its impact on mask cost and design performance simultaneously. Gupta et.al. [146] propose such an approach for minimizing mask cost by relaxing OPC tolerances on standard cells, subject to meeting timing constraints . In this flow, OPC tolerance optimization is performed

prior to tapeout.

Characterization of mask cost and timing impact of different OPC tolerances is the basic step for a complete *design-aware* OPC flow. In this work, we characterize mask cost of standard cells and wires without performing OPC repeatedly with different tolerances. Based on the statistical analysis of fractured polygon count (FC) of standard cell and wire patterns, we construct models and look-up tables of mask cost that can be used for OPC tolerance optimization. For standard cells, we give models of mask cost of polysilicon layer with inner tolerance (IT), outer tolerance (OT), starting side (SSIDE) and fragmentation edgelength (FRAG) of feature edges in the layout. Design engineers can perform trade-offs between parametric yield and mask cost using this model. RET engineers can use the model of mask cost to tune their OPC recipes without running OPC and fracturing. Since the model provides mask cost as a function of layout parameters, library designers can modify device layout to minimize mask cost without running OPC and fracturing repeatedly. Further, mask cost characterization (MCC) can be used to drive mask-friendly layout optimizations that can potentially improve yield.

OPC adjusts edge placement of features in the layout according to tolerance combination within the specified number of iterations. In addition to tolerance combination, the final fracture count of an OPC'ed layout depends on the convergence criterion of the OPC algorithm and the edgelength of fragments. Since OPC algorithm is iterative, modeling edge placement of features and fracture count is very difficult. Instead, we model the response of the OPC algorithm as a function of OPC tolerances and layout parameters. Layout dimensions and geometries of devices in standard cells is different from that of wires. Hence, we perform MCC for standard cells and wires differently.

To build mask cost models, we assume that fracture counts are generated with sign-off OPC recipes and optical models. Unless otherwise mentioned, fracture count of a standard cell refers to the fracture count of polysilicon (poly) layer only. In this work, we do not consider assist feature insertion during MCC.

VI.B.1 Library MCC (LMCC)

Fracture count of a standard cell is a function of OPC tolerances (IT, OT, SSIDE, FRAG) and its layout context. In the absence of any layout feature within

the optical radius of influence, fracture count depends entirely on IT, OT, SSIDE and FRAG. We refer to the absence of features within the distance of optical radius as isolated context. MCC for isolated context can be performed by running OPC followed by fracturing on individual standard cells for different IT, OT, SSIDE and FRAG. But in the presence of other standard cells, MCC with IT, OT, SSIDE and FRAG variation is CPU intensive. In real layouts, standard cells exist in many different layout contexts with other standard cells. Apart from the different types of standard cells surrounding a given cell, the placement of cells within the optical radius also impacts the fracture count. Running OPC and fracturing on all possible contexts with different spacing between standard cells is practically infeasible. To characterize mask cost of standard cells in a real layout context, we first identify different layout parameters that impact fracture count in the isolated context. In this work, we perform MCC for isolated context only.

Fracture count of poly layer of a standard cell in isolated context varies primarily with IT, OT, SSIDE and FRAG. Inner tolerance (IT) specifies the maximum tolerable edge movement *inside* the drawn feature. Outer tolerance (OT) specifies the maximum tolerable edge movement *outside* the drawn feature. Starting side (SSIDE) provides offset distances (inside and outside a drawn edge) that can be used by the OPC tool to converge on the edge movements faster. Fragmentation (FRAG) is one of the parameters that can have a significant impact on fracture count. OPC tools fragment a layout feature into segments and perform movement of these segments to correct the feature. The number of segments and hence the number of edge movements depend on the granularity of fragmentation. Fracture count of poly is inversely proportional to the fragment edgelength. Fine-grained fragmentation may result in fine-grained edge movements, that improve image quality. However, fracture count increases rapidly as maximum fragment edgelength is decreased. Variation of fracture count for different IT and OT for different fragmentation edgelengths is shown in Figure VI.9. For any given IT (or OT), we can observe a decrease in fracture count as the fragment edgelength is increased. In addition to the parameters described above, fracture count also depends on OPC corrections performed at line-ends and corners. To keep our exploration space limited, we do not vary line-end and corner correction parameters.

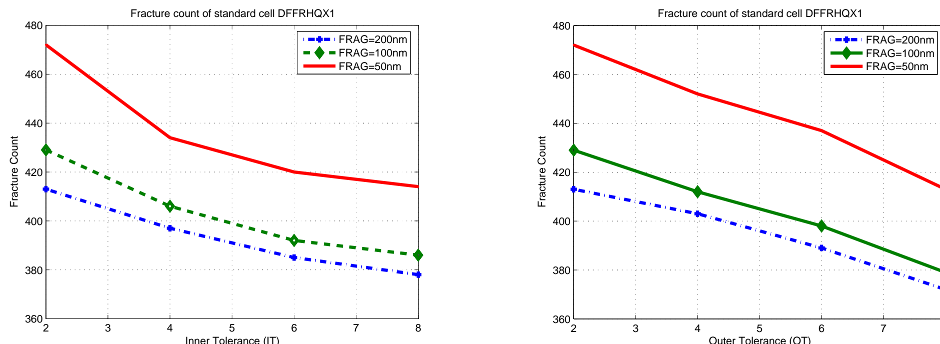


Figure VI.9: Fracture count variation with IT and OT for maximum fragment edge-lengths of 50nm, 100 nm and 200nm.

Figure VI.10 shows the flow chart for building MCC model for standard-cell library. To construct a FC model, we choose a subset of standard cells from the library and run OPC exhaustively on all combinations of IT, OT, SSIDE and FRAG. We then perform fracturing on all OPC stream files and collect fracture count data. We identify layout parameters that are the source of fracture count variation between any two standard cells for a given tolerance combination. We then perform linear regression to fit the fracture count of standard cells as a function of layout parameters for each tolerance combination.

Layout Parameter Extraction

In this part of the chapter, we explore different characteristics of standard-cell layouts that are the source of variation in fracture count for a given tolerance combination. Figure VI.11 shows poly and active layers of two standard cells. The poly fracture counts of these two cells for any given tolerance combination. The source of fracture count discrepancy is the layout of the cells. From an initial observation, we can notice that the two cells differ in number of poly features (NP), cell width (CW) and spacing between poly (PS). The actual fracture count depends on optical interactions between the features, which in turn depends on the distribution of spacing between the features. Capturing the distribution of spacing between the poly increases the complexity of analysis. In this work, we only consider the average spacing between poly, defined as the ratio

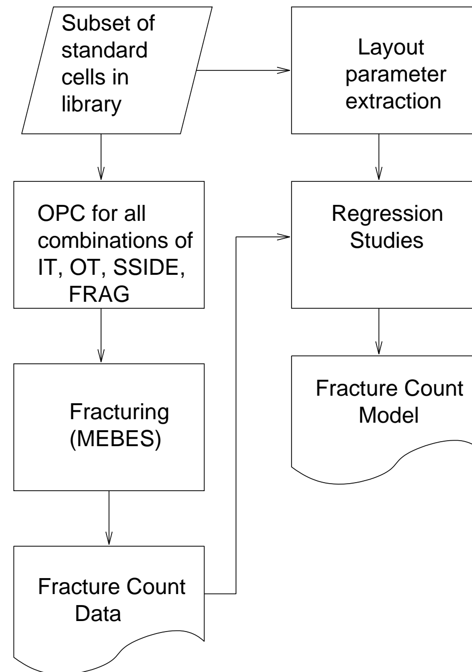


Figure VI.10: Flow chart of LMCC methodology.

of cell width to the number of poly features. The parameter NP does not differentiate between a "simple" vertical poly and a "fingered" poly with parallel devices. To capture the complexity of features, we consider poly perimeter (PW), which is the total perimeter of poly in the standard cell. To capture the impact of line ends and corners, we consider poly vertex count (PVC) which is the total number of vertices of all poly features in the standard cell. A simple vertical poly has just four vertices. If a notch is added to the vertical poly, the vertex count increases to eight, reflecting the addition of two convex corners and two concave corners.

Experimental Setup

We now give details of our OPC setup and regression studies. To build the fracture count model, we choose 15 of the most frequently used cells from standard-cell benchmarks in the 90 nm technology. We construct optical model with the parameters given in Table VI.5. We construct OPC recipes to run OPC exhaustively on all 15 cells for all combinations of IT, OT, SSIDE and FRAG given in Table VI.6. We use CalibreOPC [162] to run OPC and FractureM (MEBES) to compute total fracture count.

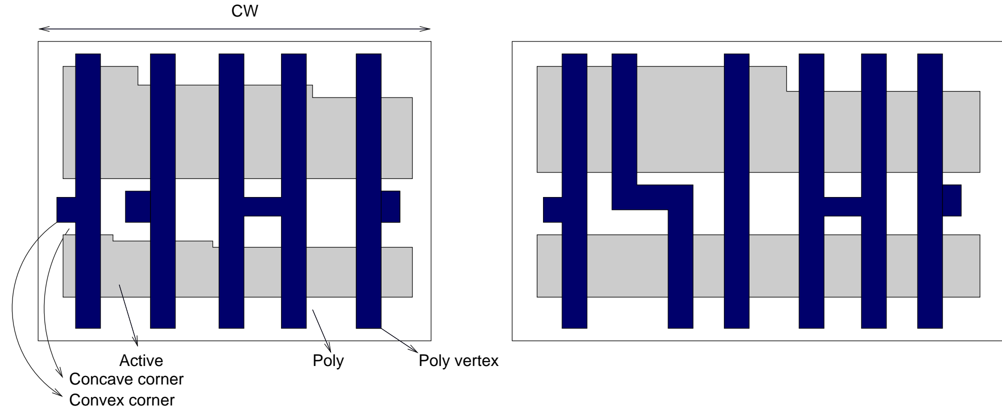


Figure VI.11: Poly and active regions of two standard cells.

Table VI.5: Optical model parameters.

λ	0.193
NA	0.68
σ_1, σ_2	0.85, 0.57
Defocus	-0.135
Illumination	Annular
Reference threshold	0.3

Table VI.6: OPC parameters.

IT	$\{-2, -4, -6, -8\}$nm
OT	$\{2, 4, 6, 8\}$nm
SSIDE	$\{-20, -10, 0, 10, 25\}$nm
FRAG	$\{50, 100, 200, 500\}$nm

IT and OT are implemented using `epeToleranceTag`. SSIDE is implemented using `opcTag -hintOffset` and FRAG using `maxedgelength` parameter in the fragmentation algorithm. Layout parameters outlined in Section VI.B.1 are extracted by analyzing standard-cell GDSII.

Based on the fracture count data and layout parameter values for 15 cells, we perform regression studies to construct FC model using SPLUS [150] software. Figure VI.12 shows a pair-wise scatter plot of poly fracture count (PFC) along with layout parameters. Column 1 of the figure shows the trend in PFC with NP, CW, PS, PVC and PW. From the plots we can observe that PFC is strongly correlated with NP, CW, PVC and PW. Since PW is strongly correlated with NP, we choose one of these two for regression.

For each tolerance combination, we run linear regression to fit FC of all 15 cells as a function of layout parameters. Figure VI.13 shows the response and the fit for all 15 cells for a single tolerance combination (IT = -6, OT = 6, SSIDE = 0, FRAG = 50).

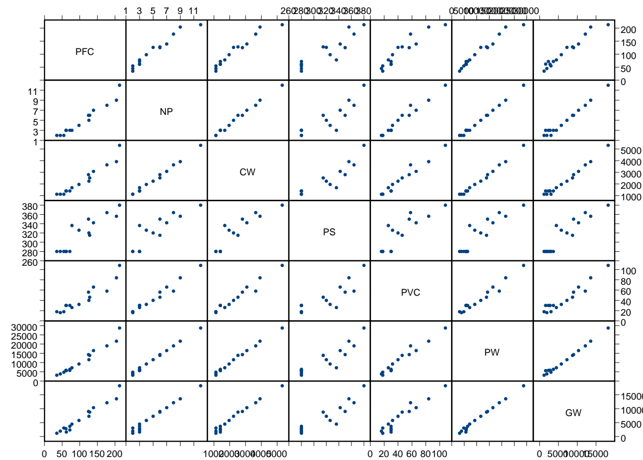


Figure VI.12: Pair-wise scatter plot showing trends between poly fracture count (PFC) and layout parameters and between different layout parameters.

Average variance of fit for all 80 tolerance combinations is 0.96, which implies that 96% of FC variation is accounted for using NP, PVC and PW. To test the fidelity of the fit, we predict FC of 100 cells using the 15-cell model. We compare predicted FC values to actual FC numbers obtained from OPC and fracturing. Figure VI.14 shows predicted and actual FC of 100 cells. From the results we observe that around 62% of the cells have less than 5% error between predicted FC and actual FC. Around 77% of the cells have less than 10% error. For all the remaining cells, the trend in predicted FC tracks that of actual FC closely. Even though the predicted FC differs from the actual FC, the trend in FC is useful for performing tolerance optimization, since the designer needs to be aware of the change in FC rather than the absolute value.

VI.B.2 Wire MCC (WMCC)

Wire mask cost (WMC) model predicts the FC of wires before running OPC using pre-characterized models and look-up tables. The objectives of WMCC are: (a) to estimate the change of FC due to different OPC tolerances and (b) to predict total FC of a layout prior to OPC and fracturing. WMC model can be used to guide choice of OPC tolerances for different regions of a metal layer. In addition to tolerance optimization, WMC model can be used for guiding wire sizing optimizations to minimize FC.

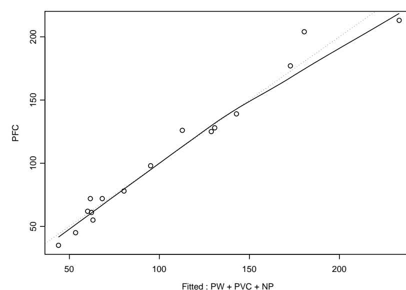


Figure VI.13: Scatter plot showing actual PFC (dots) versus fit (line) based on three variables, NP, PVC and PW.

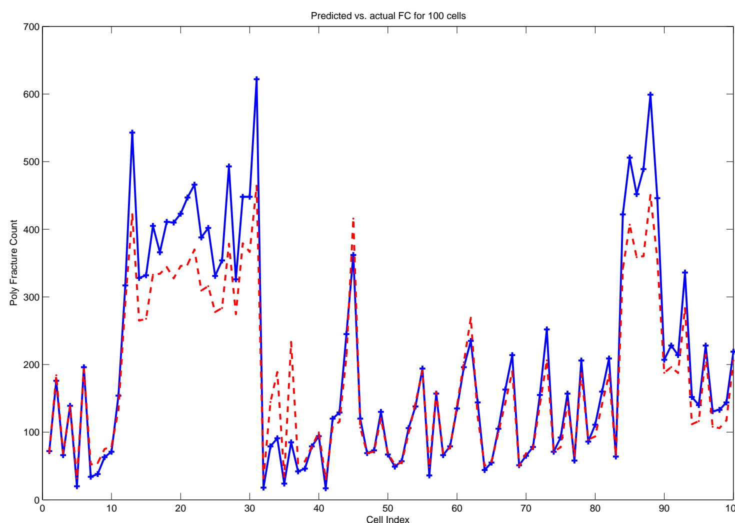


Figure VI.14: Predicted vs. actual FC of 100 standard cells. The prediction is based on model built using 15 standard cells only.

WMC model is a combination of closed-from equations and look-up tables (LUTs) to estimate FC of wires in a layout. The model captures the three major contexts of a wire such as the line-body (L), line-end (LE) and the line-corner (C) as shown in Figure VI.15. To construct a model for WMC, we characterize the response of each of the different parts of a wire pattern in various layout contexts for different OPC tolerances. The first step in WMC is the construction of different wire patterns that vary layout parameters of interest. In the next step, we run OPC and fracturing on the wire

patterns and collect FC data. Using the FC data, we construct closed-form expressions and populate LUTs representing the model. To classify a wire segment as a line-body, a line-end or a line-corner, we first study how FC varies for each part. We introduce a new concept called FC saturation that helps in this classification. Section VI.B.2 describes FC saturation in detail. Section VI.B.2 describes WMC model construction and validation for each context of a wire in detail.

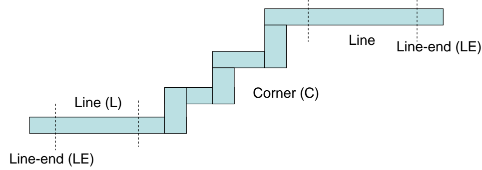


Figure VI.15: An example of context with line-body, line-end and corner shapes.

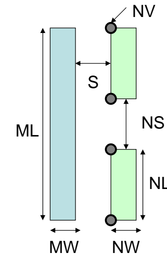


Figure VI.16: An example of context for WMC glossary.

FC Saturation

The geometries of wires in the layout along with context can be very complex. To model FC of wires in real layout contexts, we analyze the impact of increasing the number of neighboring wires to a main wire. Figure VI.16 shows a context of main pattern with two small neighbors. Parameters for the main pattern start with the letter 'M' and those of the neighbors start with the letter 'N'. A glossary of different parameters of the pattern in Figure VI.16 are summarized in Table VI.7.

Table VI.7: WMC glossary.

Parameter name	Description
ML	Main (Primary) pattern length
MW	Main (Primary) pattern width
NL	Neighboring pattern length
NW	Neighboring pattern width
NV	Vertex of neighboring pattern
S	Distance between the main pattern and the neighboring pattern
NS	Distance between line-ends of neighboring patterns

FC of the main pattern typically decreases as S, NS and NL increase. As S increases, the impact of top right neighbor on OPC of the main pattern decreases. As NS increases, the combined impact of the neighbors on the main pattern decreases. It is interesting to note that as NL increases, FC of the main pattern decreases. The diffraction effects caused by a small neighbor in the vicinity of the main pattern are corrected aggressively by the OPC tool. As NL increases, the diffraction effects spread across the entire length of the main pattern, which are corrected uniformly across the entire length of main pattern. This results in a smaller impact on FC of the main pattern. The decrease in FC of the main pattern ceases as S, NS and NL increase beyond a certain limit. This phenomenon is referred to as saturation and the distance at which saturation takes place is called the saturation point of that parameter. The saturation point of each parameter is identified by a subscript 'o' to the parameter. Saturation point is experimentally determined by varying a layout dimension and measuring the corresponding FC. From our experimental results, we observe that the saturation point of S (i.e., S_o) is equal to the optical radius (OR) of the model used for OPC. The parameters S_o, NS_o and NL_o specify the saturation point of S, NS and NL respectively. The OPC treatment of line-ends and corners of a wire pattern is different from that of the line-body. To determine the extent over which OPC treatment of a line-end or a corner impacts that of a line-body, we construct test patterns shown in Figure VI.17. We define two parameters, line-end characteristic length (LECL) and corner characteristic length (CCL) to quantify the impact of line-ends and corners respectively. The saturation points of these parameters are $LECL_o$ and CCL_o. The general trend in saturation points of various parameters are shown in Figure VI.18 and their values are summarized in Table VI.8. From the results, we observe the following relationships between saturation parameters and OR. We verified these across a different set of OPC recipe and optical model.

$$S_o, LECL_o, CCL_o = OR \tag{VI.2}$$

$$NS_o, NL_o = 2 \times OR \tag{VI.3}$$

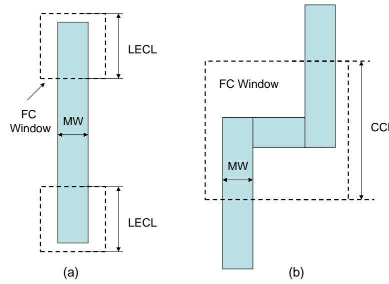


Figure VI.17: Test structures for (a) line-end characteristic length (LECL) and (b) corner characteristic length (CCL).

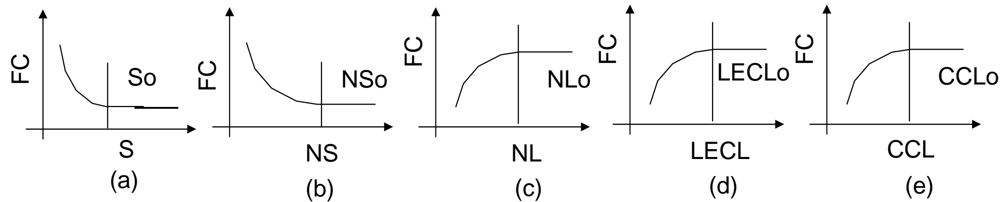


Figure VI.18: General trend in FC saturation with (a) S, (b) NS, (c) NL, (d) LECL and (e) CCL.

Line-body, End and Corner Models

Based on the saturation points for different parameters, we construct different configurations of the line-body, line-end and line-corners and run OPC followed by fracturing. To analyze FC for each type of pattern, we define a parameter, FC window, that determines the region over which FC is measured. Size of FC window is determined by saturation points.

Table VI.8: Saturation point characteristics: $S_0 = 0.65\mu$m, $NS_0 = 0.13\mu$m, $NL_0 = 0.13\mu$m, $LECL_0 = 0.65\mu$m and $CCL_0 = 0.65\mu$m.

Dimension (S/NS/NL) (μm)	0.3	0.65	1.3	1.5
FC for S	101	79	79	79
FC for NS	158	138	124	124
FC for NL	86	93	101	101
FC for LECL	47	59	60	60
FC for CCL	69	71	71	71

For the line-body pattern, we observe that the FC is a first degree polynomial function of ML with a slope α and intercept β (FC = αML + β(S,EPE_{tol})) where, EPE_{tol} is the EPE tolerance specification of the main line. The slope of the function is independent of pattern parameters and is dependent only on the optical parameters. In the presence of neighbors, we observe a shift in the FC of the line-body and this is captured by β. The intercept itself is a function of spacing between the main line and the neighbor, and EPE tolerance. To compute the slope of the model, we construct the test patterns shown in Figure VI.20(a) which shows two main lines M1 and M2 of lengths ML_1 and ML_2 respectively. The value of α is computed as $FC(M2) - FC(M1)/ML_2 - ML_1$. To obtain the value of β we construct LUT with S and EPE_{tol} as parameters. The test pattern shown in Figure VI.20(b) is constructed for different values of S and OPC is run at different EPE_{tol}. To verify the model, we compute α and generate LUT for β and predict FC of two test patterns as shown in Figure VI.19. The LUT for β with S and EPE_{tol} and the comparison of predicted FC with experimental results for two examples are summarized in Table VI.9. FC increase with increase in number of neighbors is additive. E.g., FC of main line in Figure VI.19(b) is $\alpha\text{ML} + \beta(S1, EPE_{tol}) + \beta(S2, EPE_{tol})$.

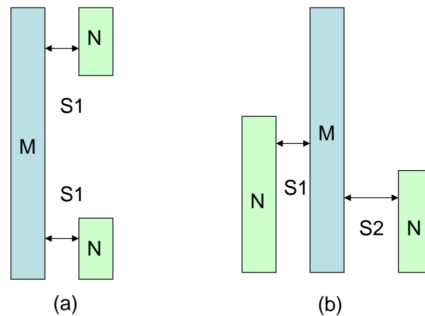


Figure VI.19: Two examples for validation of line-body model: (a) two neighbors with same space ($S1 = 200nm$) and (b) two neighbors with different spaces of ($S1 = 200nm$ and $S2 = 400nm$).

For the line-end model, we consider two types of patterns as shown in Figure VI.21. Part (a) of the figure shows a line-end with a single neighbor and part (b) shows the line-end with two neighbors at equal spacings from the main line. FC model for line-end is based completely on LUTs. For a line-end with single neighbor, we construct

Table VI.9: LUT for line-body model and comparison of FC with simulation and experimental results for two examples: maximum difference of FC is 5.

EPE_{tol}	**LUT (Space: nm)**			**FC of example (a)**		**FC of example (b)**	
	200	400	600	Experiment	Simulation	Experiments	Simulation
0.002	23	10	3	123	124	114	111
0.003	24	12	0	127	126	109	114
0.005	15	6	0	109	108	101	99
0.010	13	6	0	103	104	98	97

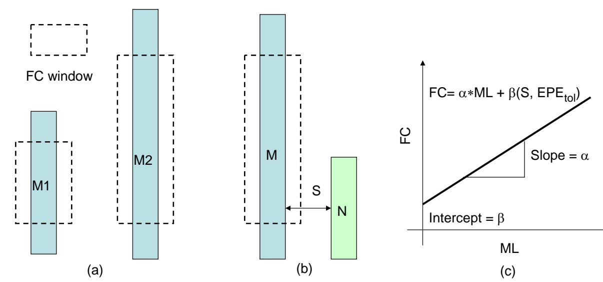


Figure VI.20: Line-body model: (a) Test patterns for α calculation, (b) test patterns for β LUTs generation and (c) line-body model ($FC = \alpha\text{ML} + \beta(S, EPE_{tol})$).

the LUT by varying S and EPE_{tol}. For a line-end with two neighbors, we observe that FC variation is a function of spacing to the smaller of the two neighbors. To validate the line-end model, we construct test patterns shown in Figure VI.22. FC window for the line-end is determined by value of $LECL_o$. Table VI.10 gives the LUTs for line-end for single and double neighbor cases. Table VI.11 compares the predicted versus measured FC for line-ends shown in Figure VI.22 for different neighbor spacings. If a line-end with asymmetrically-spaced neighbors is encountered in a real layout, we choose the neighbor that is closer to the main line. Line-ends are fragmented heavily during OPC. The presence of wire patterns around the line-end does not change the fragmentation significantly and hence, we see a small change in FC with change in neighbor spacing.

WMC model for the line-corner is generated by running OPC and fracturing on line-corner patterns with different CCL and EPE_{tol}. Based on the analysis of real layouts, we observe that the number of line-corners within the FC window (determined

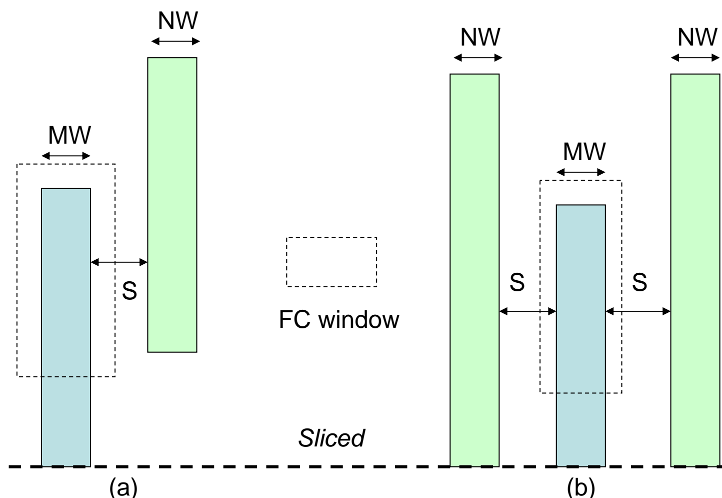


Figure VI.21: Pattern examples for line-end model: (a) line-end with single neighbor and (b) line-end with double neighbors.

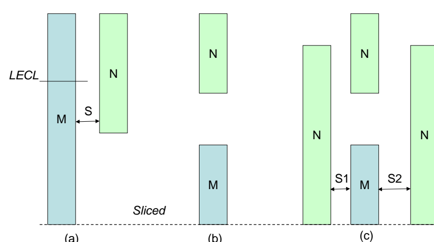


Figure VI.22: Three examples for line-end model validation: (a) line-end with S = 200nm, (b) line-end perpendicular neighbor, (c) line-end with S1 = 200nm and S2 = 400nm.

by CCL_o) does not exceed two. Hence, we construct LUTs with CCL and EPE_{tol} as parameters and use it for predicting FC of line-corners in real layouts.

FC Prediction

To validate the line-body, line-end and line-corner WMC models presented above, we analyze a real layout and predict its FC and compare it with real FC after OPC and fracturing. For each metal layer, the model generator constructs a test layout that captures the three contexts of a wire pattern. We then extract the optical radius (OR) of the OPC model from a line and space test pattern. We then run OPC and

Table VI.10: LUTs for single and double neighbors cases of the line-end model: $LECL_0$ is 650nm.

EPE_{tol}	**Single neighbor (Space: nm)**				**Double neighbor (Space:nm)**			
	200	400	600	650	200	400	600	650
0.002	16	15	13	12	14	13	12	12
0.003	17	14	13	12	13	12	11	12
0.005	15	14	11	11	12	12	11	11
0.01	12	10	10	9	10	9	9	9

Table VI.11: Comparison of predicted FC with real FC for the three line-end test patterns in Figure VI.22.

EPE_{tol}	**FC for example (a)**		**FC for example (b)**		**FC for example (c)**	
	Experiment	Simulation	Experiment	Simulation	Experiment	Simulation
0.002	15	16	12	12	13	14
0.003	14	17	11	12	12	13
0.005	13	15	11	11	11	12
0.010	11	12	10	9	10	10

fracturing on the test layout and record FCs of contexts. These FCs are used to compute α and populate LUTs.

To predict FC of real layouts, the predictor decomposes wire patterns from real layouts into the three contexts based on FC windows. FCs for patterns in each category are computed using the models and LUTs. Total FC of the layout is the sum of FC for each context. Table VI.12 shows the real and predicted FC for metal layer 2 of ALU128 benchmark in the 90 nm technology. The maximum error in prediction is $\pm 6\%$.

Table VI.12: Real versus predicted FC for different EPE_{tol} of metal layer 2 from ALU128 benchmark implemented in the 90 nm technology.

EPE_{tol} **(nm)**	**Real FC**	**Predicted FC**	**Error(%)**
0.002	306234	297022	3
0.004	282202	265312	6
0.006	227529	242100	-6
0.008	213306	221046	4.5

VI.B.3 Conclusions

In this section of the chapter, we have presented methodologies for characterizing OPC of standard cells and wire patterns in terms of fracture count. The characterization provides models of FC as a function of layout parameters and OPC tolerances. FC model constructed for library MCC using a limited set of library cells for a given tolerance combination can predict the FC trend of up to 75% of cells in the library within 5% error. FC model for wires can predict actual FC of layouts within 6% error. These FC models can be used by designers to choose between different OPC tolerance combinations to minimize mask cost. RET engineers can extend the presented models by adding other OPC parameters and use it for tuning OPC recipes and optical model parameters. Library designers can use these models for constructing design rules that minimize mask cost without actually running OPC and fracturing. The placement of standard cells and the type of standard cells surrounding a given cell have significant impact on its FC. We are currently working on extending the library MCC approach to include the impact of layout context. Line-ends and corners of poly features contribute significantly to FC. Optimizing line-end corner fragmentation parameters can enable further reductions in FC.

VI.C Acknowledgments

Chapter VI is in part a reprint of “Modeling OPC Complexity for Design for Manufacturability”, *Proc. BACUS Symposium on Photomask Technology and Management*, 2005 and “Performance-Driven OPC for Mask Cost Reduction”, submitted to *SPIE Journal of Microlithography, Microfabrication and Microsystems*. I would like to thank my coauthors Chul-Hong Park, Swamy Muddu, Dr. O. Sam Nakagawa, Jie Yang, Dr. Dennis Sylvester and Dr. Andrew B. Kahng.

VII

Coping with BEOL Variability: Performance-Aware Metal Fill Synthesis

VLSI technology has entered deep submicron regimes, where the manufacturing process increasingly constrains physical layout design and verification. *Chemical-mechanical planarization* (CMP) [135, 137] and other manufacturing steps in nanometer-scale VLSI processes have varying effects on device and interconnect features, depending on local attributes of the layout. Uniformity of CMP, which is used for planarization of interlayer dielectrics (or oxide, with newer shallow-trench isolation) in multi-layer i¿nterconnect processes, depends on uniformity of features on the interconnect layer beneath a given dielectric layer to avoid dishing and other irregularities. To improve manufacturability and performance predictability, foundry rules require that a layout be made uniform with respect to prescribed density criteria, through insertion of *area fill* ("dummy") geometries.[1]

All existing methods for synthesis of area fill are based on discretization [126, 127]: the layout is partitioned into *tiles*, and filling constraints or objectives (e.g., minimizing the maximum variation in feature area content) are enforced for square *windows* that each consists of $r \times r$ tiles. In practice, then, layout density control is achieved by

[1] For example, a local interconnect metal layer might have a requirement that every $10\mu m \times 10\mu m$ window contain at least 35 μm^2, but no more than 70 μm^2, of metal features [124, 140].

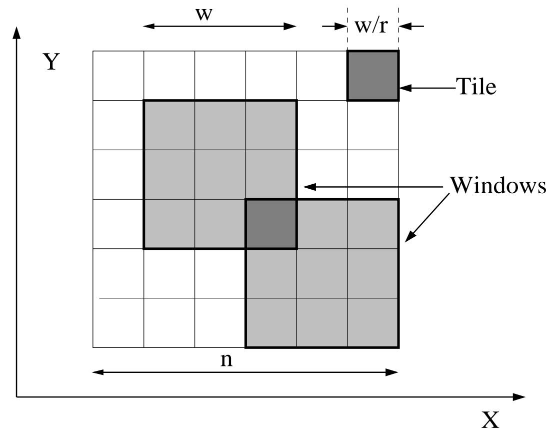


Figure VII.1: In the fixed r-dissection framework, the n-by-n layout is partitioned by r^2 (here, $r = 3$) distinct overlapping dissections with window size $w \times w$. This induces $\frac{nr}{w} \times \frac{nr}{w}$ *tiles*. Each dark-bordered $w \times w$ window consists of r^2 tiles.

enforcing density bounds in a finite set of windows. Invoking terminology from previous literature, we say that the foundry rules and EDA tools (physical verification and layout) attempt to enforce density bounds within r^2 overlapping *fixed dissections*, where r determines the amount w/r by which the dissections are offset from each other. The resulting *fixed r-dissection* (see Figure VII.1) partitions the $n \times n$ layout into tiles T_{ij}, then covers the layout by $w \times w$-windows W_{ij}, $i, j = 1, \ldots, \frac{nr}{w} - 1$, such that each window W_{ij} consists of r^2 tiles T_{kl}, $k = i, \ldots, i + r - 1$, $l = j, \ldots, j + r - 1$.

While area fill feature insertion can significantly reduce layout density variation, it can also change interconnect signal delay and crosstalk by changing coupling capacitance. These changes can be harmful to timing closure flows, especially since fill is typically added as a physical verification or even post-GDSII (at the foundry) step. Therefore, in addition to satisfying density requirements, dummy fill insertion should also minimize *performance impact*. However, the issues associated with capacitance and area fill are complex and there is no existing published work on performance-driven area fill synthesis.[2]

Our contributions include:

[2]Although this concept has been recently mentioned in some startup web sites [139, 143, 144], no details of functionality are given. Currently, metrological methodologies are used to determine the "best" choice of buffer distance, dummy fill type (grounded versus floating), and dummy fill pattern.

- new *PIL-Fill (Performance Impact Limited Fill)* problem formulations for performance-impact limited area fill synthesis;
- a practical integer linear programming formulation, along with a greedy method, for the Minimum Delay, Fill-Constrained variant of the PIL-Fill problem;
- an iterated greedy method between area fill synthesis and static timing analysis for the Maximized Minimum Slack, Fill-Constrained variant of the PIL-Fill problem;
- experimental results of the two MDFC PIL-Fill methods, confirming the advantages of our work over previous methods (up to 90% reductions in delay impact compared with normal fill methods [126]), and identifying at least one practical method for deployment; and
- experimental results of the iterated greedy method for MSFC PIL-Fill problem, showing significant advantage with respect to maximizing the minimum slack over all nets in the post-fill

Fill can be either grounded or floating. Grounded fill has the advantage of having predictable parasitic impact but suffers from huge routing overheads. Floating fill is tougher to extract but has little layout impact. Our present work assumes that area fill consists of squares of floating fill; we seek a fill placement with minimum delay impact of fill insertion. In the next section, we review related works in the PIL-Fill domain. In Section VII.B, we briefly review interconnect capacitance estimation models, and describe our simplified capacitance impact and delay impact model for floating fill. Section VII.C formulates the PIL-Fill problem with two different objectives, and solution approaches are given in Section VII.E and Section VII.F. Section VII.G gives experimental results and we conclude in Section VII.H.

VII.A Related Work

According to Stine et al. [141], to minimize the increase in interconnect capacitance that results from area fill, (i) the total amount of added fill should be minimized, (ii) the linewidth of the fill pattern should be minimized, (iii) the spacing between fill lines

should be maximized, and (iv) the buffer distance should be maximized. Unfortunately, these guidelines are rather generic. We observe that restricting the amount of dummy fill and increasing the buffer distance has the unwanted effect of limiting the possible improvements in uniformity achieved by fill insertion. Furthermore, such guidelines are not precisely matched to the relevant underlying criteria, e.g., capacitance minimization does not comprehend the delay and timing slack impact of added capacitance. While no work has (to best of our knowledge) yet addressed the PIL-Fill problem, two related works are of interest.

Work at Motorola by Grobman et al. [133] points out that the main parameters to influence the change in interconnect capacitance due to fill insertion are feature ("block") sizes and proximity to interconnect lines. The larger the size of the block, the larger the consequent interaction between interconnect lines. Similarly, the closer blocks are to interconnect lines, the stronger their interaction will be. They also consider several structures that are expected to represent the most profound effects. In one limiting case, dense lines effectively do not suffer much from floating fill placement above and below, since their capacitance is dominated by coupling to neighbors. On the other hand, when interconnect lines are more sparsely situated, floating fill has greater performance impact. The importance of dummy fill size is also examined: large block shapes more effectively transmit local effects to their extent, and hence if filling is to be performed over critical paths, use of smaller fill blocks with the same filling density helps limit the increase of interconnect capacitance.

Work at MIT Microsystems Technology Laboratories [141] proposes a rule-based area fill methodology. To minimize the added interconnect capacitance resulting from fill, a dummy fill design rule is found by modeling the effects on interconnect capacitance of different design rules (which are consistent with the fill pattern density requirement). Three canonical parameters are considered in design rules: buffer distance (buf), block width (w), the block space (s). Effects of fill on interconnect capacitance are calculated from the canonical parameters as well as line width and spacing. The calculation result is coupled with the minimum pattern density goals to obtain an optimized dummy fill design rule. It is important to note that the MIT methodology yields only a *rule*: the fill insertion is not driven by any context (e.g., per-net or per-wire segment

delay or slack considerations).

Lee et al. [136] describe the methodology used at Samsung for chip-level metal fill modeling. Their approach replaces the metal fill layer by an effective (i.e., equivalent) high-k dielectric. The increments of capacitance due to floating metal fill are dependent on the signal line width and spacing, inter-metal dielectric thickness and permittivity, density of metal fills, metal fill feature size, and metal layer thickness. RC extraction results in [136] show that the total interconnect capacitance increase can be up to 15% for some nets in an $0.18\mu m$ design. Thus, floating dummy metal fills should be included in chip-level RC extraction and timing analysis to avoid timing errors.

VII.B Capacitance and Delay Models

Works on multi-layer interconnect capacitance extraction include 1-D, 2-D, 2.5-D and 3-D analytic models [122, 123, 130, 131, 142]. In general, the capacitance of interest at any node consists of three components: (i) *overlap (area) capacitance*, C_a, formed by the surface overlap (in two dimensions) of two conductors; (ii) *lateral coupling capacitance*, C_{lt}, between two parallel conductors on the same plane; and (iii) *fringe capacitance*, C_{fr}, that represents coupling between two conductors on different planes. In other words, the interconnect capacitance at any node is given by

$$C_t = C_a + C_{lt} + C_{fr} \quad \text{(VII.1)}$$

The effect of small floating dummy features on overlap and fringing capacitance of active (switching) lines is smaller than that on lateral coupling capacitance [122]. Fringing capacitance can be accounted for by more complicated capacitance models while taking overlap capacitance into consideration would require cognizance of more than one layer at a time. We only consider the impact of area fill on the lateral coupling capacitance between active lines.

A typical fill insertion approach is to grid the layout into sites according to the fill feature size and design rules, then insert the fill features into the empty sites to satisfy the density requirements. To estimate area fill impact on active line delay, we focus on the capacitance increment in the active line due to the fill. In Figure VII.2(A), the total

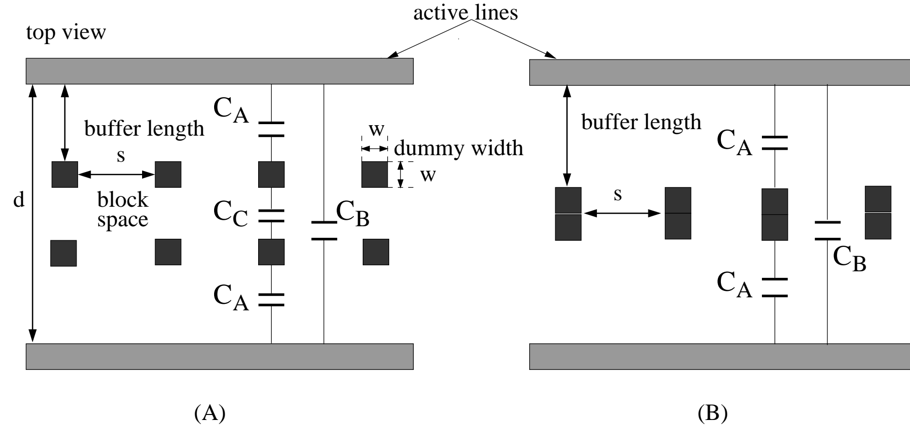


Figure VII.2: Example configurations of floating dummy fill.

capacitance of an active line before area fill is inserted can be written as

$$C_{orig} = C_B \cdot l = \frac{\epsilon_0 \epsilon_r a}{d} \cdot l \tag{VII.2}$$

where C_B is the per-unit length capacitance between the active line and its neighboring active line, l is the overlap length of the two active lines, ϵ_0 is permittivity of free space, ϵ_r is the relative permittivity of the material between the two conductors, d is the distance between the two active lines and a is the overlapping area between them.

For the general case (with two rows of dummy fills) in Figure VII.2(A), the total capacitance between two active lines is

$$C_{fill} = (\frac{1}{1/C_A + 1/C_C + 1/C_A}) \cdot w \cdot k + C_B \cdot (l - w \cdot k) \tag{VII.3}$$

where C_A is the capacitance between the dummy feature and the active line, and C_C is the capacitance between the dummy features. In this equation, w is the dummy feature width, s is the space between dummy features, and k is the number of dummy features between the two active lines. We assume that the floating dummy features have no effect on C_B.[3] To simplify the estimation, we use a simple parallel plate capacitance model. We can then approximate the impact of two rows of dummy features by making one combined row of dummy features, as shown in Figure VII.2(B). Generalizing to m rows

[3] In the parallel plate capacitance model that we have assumed, C_b is not affected. In reality there will be some effect on C_b due to fringing.

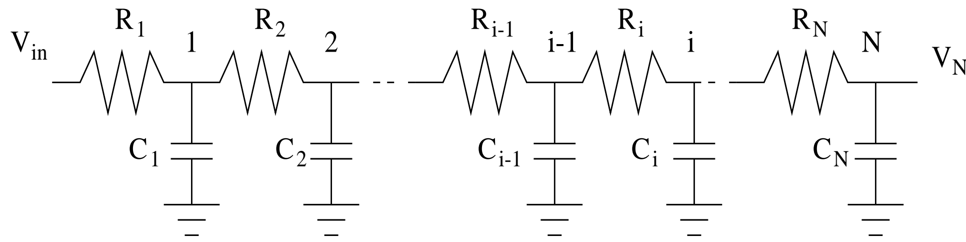


Figure VII.3: Segmented RC line model.

of dummy features, we obtain the following estimate of per-unit coupling capacitance between two active lines separated by a column of m dummy features:

$$C_{A'} = f(m) = \frac{\epsilon_0 \cdot \epsilon_r \cdot a}{d - m \cdot w} \tag{VII.4}$$

$$\approx \epsilon_0 \cdot \epsilon_r \cdot a \cdot (1 + \frac{m \cdot w}{d})/d$$

$$= C_B + \frac{\epsilon_0 \cdot \epsilon_r \cdot a \cdot m \cdot w}{d^2} \tag{VII.5}$$

Here, we model m dummy features in one column between two active lines as a single metal block with length $m \cdot w$ between the lines. When $w \ll d$, we can further simplify the calculation as a linear one (see Equation (VII.5)), where $\frac{\epsilon_0 \cdot \epsilon_r \cdot a \cdot m \cdot w}{d^2}$ is the incremental capacitance due to dummy feature insertion. Then, the total capacitance between two active lines can be estimated as

$$C_{fill} = C_{A'} \cdot w \cdot k + C_B \cdot (l - w \cdot k). \tag{VII.6}$$

With respect to interconnect delay, our discussion below will use the Elmore delay model to estimate total delay increase due to area fill. Elmore delay [132] of a cascaded N-stage RC chain (Figure VII.3) as

$$\tau_N = \sum_{i=1}^{N} R_i \sum_{j=i}^{N} C_j = \sum_{i=1}^{N} C_i \sum_{j=1}^{i} R_j \tag{VII.7}$$

Each node j on the chain contributes to τ_N, the product of the capacitance at node j and the total resistance between j and the source node. If the capacitance at node i increases by ΔC_i, the increment of Elmore delay at any node k below node i is

$$\Delta\tau_k = \Delta C_i \sum_{j=1}^{i} R_j \tag{VII.8}$$

Equation (VII.7) implies that Elmore delay is an additive with respect to capacitance along any source-sink path. That is, if we add the coupling capacitance C_x at position x, the delays at all nodes below the position x will increase by $C_x \cdot R_x$. Here, R_x is a constant, and equal to the total resistance between the source and the position x (we will refer to this as *entry resistance*, i.e., an "upstream" resistance).

VII.C Problem Formulations

Performance-impact limited area fill synthesis has two objectives:

- minimization of the layout density variation due to CMP planarization; and
- minimization of the dummy features' impact on circuit performance (e.g., signal delay and timing slack).

It is difficult to satisfy the two objectives simultaneously. Practical approaches will tend to optimize one objective while transforming the other into constraints. In this section, we propose two performance-impact limited area fill problem formulations (PIL-Fill) in which the objectives are to minimize performance impact, subject to a constraint of prescribed amounts of fill in every tile.

VII.C.1 Min-Delay-Fill-Constrained Objective

Our *minimum delay with fill constraint*, or MDFC, formulation[4], can be stated as follows [128].

Given a fixed-dissection routed layout and the design rule for floating square fill features, insert a prescribed amount of fill in each tile such that the performance impact (i.e., the total increase in wire segment delay) is minimized.

Since each tile in the fixed-dissection layout can be considered independently, we may reformulate the MDFC PIL-Fill problem on a per-tile basis. In other words, for

[4]We have also studied a *minimum variation with delay constraint* formulation, but it is less tractable to optimization heuristics and we do not discuss it here.

each tile the following optimization is separately performed.

Given tile T, a prescribed total area of fill features to be added into T, a size for each fill feature, a set of slack sites *(i.e., sites available for fill insertion) in T per the design rules for floating square fill, and the direction of current flow and the per-unit length resistance for each interconnect segment in T, insert fill features into T such that total impact on delay is minimized.*

For this per-tile MDFC PIL-Fill problem, we use the above capacitance approximations and the Elmore delay model. Under the Elmore delay model, the impact of each wire segment delay on the total sink delay of the routing net is found by multiplying by the number of downstream sinks. Thus, we define the *weight* of an active line l as

$$W_l \equiv \text{the number of downstream sinks}$$

which allows us to directly minimize total sink delay impact over all nets in a given tile.[5]

VII.C.2 Max-MinSlack-Fill-Constrained Objective

A weakness of the MDFC PIL-Fill formulation is that we minimize the total delay impact *independently* in each tile. That is, the impact due to fill features on the signal delay of complete timing paths is not directly considered. Thus, we also propose to maximize the minimum slack of all nets, still subject to a constraint of prescribed amounts of fill in every tile region of the layout. We call this a *maximum min-slack with fill constraint*, or MSFC, formulation [129].

Given a fixed-dissection routed layout and the design rule for floating square fill features, insert a prescribed *amount of fill in each tile such that the minimum slack over all nets in the layout is maximized.*

We use a commercial Static Timing Analysis tool (Cadence Pearl) to extract slacks at all pins of each net in the layout.

[5]This objective, which is correlated with total impact on sink actual arrival times, brings us closer to the ideal of being timing-slack driven.

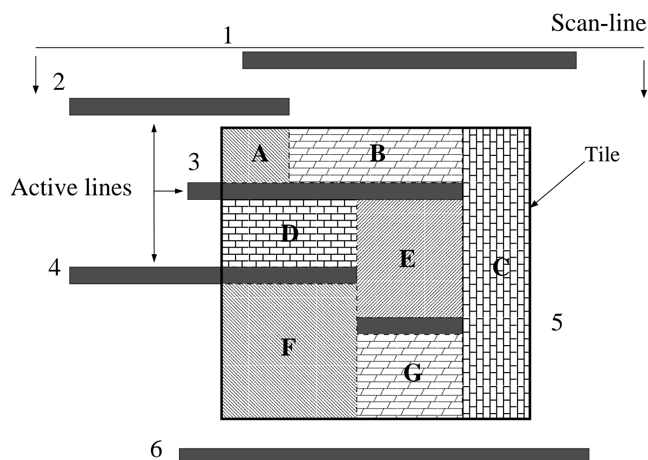


Figure VII.4: SlackColumn-III definition: Illustration of scan-line and slack blocks within tile between pairs of active lines in adjacent tiles.

VII.D Geometry Computation

The key computational geometry task in solving PIL-Fill problems is to find all pairs of parallel active line segments, as well as the slack space (i.e., empty sites where fill geometries can be inserted) between each such pair. Without loss of generality, we assume that the routing direction is horizontal on the selected layer.

We define a *fill site column* as a column of available sites for fill features between two active lines or between an active line and a layout boundary. A *slack block* is a maximal contiguous set of fill site columns having equal height (and, due to the fill site grid, equal width). Figure VII.4 shows seven such slack blocks in a tile. As an example, the fill features located in the slack block C in Figure VII.4 will affect the coupling capacitance on active lines 1 and 6. We also define the *size of a slack site column* as the number of empty sites in the column available for fill insertion.

To find such fill columns in the layout, we first obtain the position of each active line. After sorting the active lines according to y-coordinates (for horizontal routing direction) or x-coordinates (for vertical routing direction), we scan the whole layout from the bottom boundary (for horizontal routing direction) or from the left boundary (for vertical routing direction) to find the fill columns between active lines or between boundary and active line. The algorithm is described in detail in Figure VII.5.

Scan-Line Algorithm to Compute Fill Column
Input: a design-rule correct layout; tile size t; site size s **Output:** a list of fill columns $Cols$ on the layout; lists of slack columns $Cols_{ij}$ intersecting with each tile T_{ij}; lists of active lines AL_{ij} intersecting with each tile T_{ij}
1. Partition the layout into $m \times n$ tiles and $k \times l$ sites 2. Sort the list of active lines AL on layout according to its Y coordinates 3. Initialize the list of fill columns $Cols$ 4. Create k empty fill columns starting from the bottom boundary 5. **While** there is active line left in AL **Do** 6. Get the active line with smallest Y coordinate in AL 7. **For** each fill column Col_{pq} intersecting with the active line **Do** 8. **If** the size of fill column Col_{pq} is larger than 0 **Then** 9. Add Col_{pq} into $Cols$ 10. **Else** Ignore Col_{pq} 11. Create a new one starting from the active line 12. Delete active line from the list 13. **For** all fill columns Col_{pq} ended at the top boundary **Do** 14. **If** size of fill column Col_{pq} is larger than 0 **Then** 15. Add Col_{pq} into $Cols$ 16. Initialize lists of fill columns $Cols_{ij}$ and lists of active lines AL_{ij} 17. **For** each fill column Col_{pq} in $Cols$ **Do** 18. **For** each tile T_{ij} overlapping with fill column Col_{pq} **Do** 19. Calculate number of sites in Col_{pq} within the tile T_{ij} 20. Add Col_{pq} into $Cols_{ij}$ and correlated active line(s) into AL_{ij}

Figure VII.5: Scan-line algorithm to find fill slack columns on given layer (assuming horizontal routing direction).

VII.E Approaches for MDFC PIL-Fill

VII.E.1 Integer Linear Programming Approach

In our flow, we calculate post-routing interconnect slack values after obtaining routing information from a DEF file. Since (i) all dummy features have the same shape, (ii) the potential number of dummy fill features (and their positions, given the fixed-dissection layout) in each fill column is limited, (iii) the size of any fill column is also limited, and (iv) the other parameters (ϵ_o, ϵ_r, and d) in Equation (VII.4) are constant for each pair of active lines, we can pre-build a look-up table $f(n,d)$ that gives the capacitance increment for inserting n fill features between any pair of active lines that

are separated by distance d. Based on the look-up table, an accurate ILP formulation can be given. We first make the following definitions.

- $W_l \equiv$ weight of active line l;
- $C_k \equiv$ size (capacity) of feasible fill site column k for dummy features within the tile;
- $m_k \equiv$ number of dummy features inserted in column C_k;
- $Cap_k \equiv$ incremental capacitance caused by the m_k dummy features in column C_k, calculated according to Equation (VII.5);
- $\Delta\tau_l \equiv$ total delay increment on active line l due to the insertion of dummy features along it in the tile;
- $R_l \equiv$ total (upstream) resistance of path from the source node to the entry point of active line l into the tile; and
- $r_l \equiv$ per-unit resistance of active line l.
- $m_{k_n} \equiv$ auxiliary boolean variable:

$$m_{k_n} = \begin{cases} 1 & \textit{if } m_k = n \\ 0 & \textit{otherwise} \end{cases} \tag{VII.9}$$

Minimize:

$$\sum_{l=1}^{L} W_l \cdot \Delta\tau_l \qquad \text{over all active lines} \tag{VII.10}$$

Subject to:

$$F = \sum m_k \quad \text{over all fill columns} \tag{VII.11}$$

$$m_k = \sum_{n=1}^{C_k} n \cdot m_{k_n} \quad \text{for each fill column} \tag{VII.12}$$

$$\sum_{n=1}^{C_k} m_{k_n} = 1 \quad \text{for each fill column} \tag{VII.13}$$

$$Cap_k = \sum_{n=1}^{C_k} f(n, d_k) \cdot m_{k_n} \quad \text{for each fill column} \tag{VII.14}$$

$$\Delta\tau_l = \sum_k Cap_k \cdot (R_l + \sum_{s=p}^{k} r_l) \quad \text{for each active line} \tag{VII.15}$$

$$\text{Integer:} \quad 0 \leq m_k \leq C_k \tag{VII.16}$$

$$\text{Binary:} \quad m_{k_n} \tag{VII.17}$$

- The objective function (VII.10) implies that we minimize the weighted incremental Elmore delay caused by dummy feature insertions. L is the total number of active lines in the tile.

- Constraints (VII.11) ensure that the total number of *covered* (i.e.,used) fill sites is equal to the number of dummy features.

- Constraints (VII.12 and VII.13) imply that m_k can only be assigned one value from 1 to C_k.

- Constraints (VII.14) are the equations for coupling capacitance based on the look-up table. Here, $f(n, d_k)$ is the constant value from the pre-built look-up table.

- Constraints (VII.15) are used to capture the total Elmore delay increment due to dummy feature insertions in all fill columns along the active line l in the tile. $(R_l + \sum_{s=p}^{k} r_l)$ is the total resistance between the source and the position k on the active line l in the tile. p is the x-coordinate of the left-most point of the active line in the tile. k is the x-coordinate of fill column k.

- Constraints (VII.16) ensure that the number of covered fill sites in any column is no greater than the column size (capacity).

VII.E.2 Greedy Method

From Equation (VII.15), the impact on delay due to the dummy features depends on the total resistance between the source and the current node. Our final algorithmic approach for the MDFC PIL-Fill problem is to greedily insert dummy features along active line segments where the incremental delay is minimum. This greedy approach is described in Figure VII.6.[6]

VII.F Iterated Approach for MSFC PIL-Fill

To maximize minimum slack over all nets in the post-fill layout, we propose an Iterated Greedy approach based on iterations between the static timing analysis (STA) tool and the area fill synthesis. Capacitance impact due to fill feature insertion during area fill synthesis is written in Reduced Standard Parasitic Format (RSPF) as a file input to STA tool.

This approach uses the same capacitance and delay models as in the MDFC PIL-Fill approaches. After obtaining the density requirements from normal area fill synthesis and fill site columns from the scan-line algorithm, we run the industry STA tool to get the slack values of all input pins in the layout and set the slack of each active line as the minimum slack of its downstream input pins. We consider the slack value of a given fill site column to be the minimum slack of its neighboring active lines. Then, all fill site columns are sorted according to their slack values. Among them, the fill site column with maximum slack value is chosen for fill feature insertion. For each tile intersecting with this fill site column, the number of fill features actually inserted in the column is dependent on the number of required fill features of the tile, the overlapping size of the fill site column, and the column's slack value. Once a feasible number of fill features has been inserted into the tile, the number of required fill features of the tile and the size of the affected fill site column are updated. The added delay is estimated based on our capacitance and delay models, and the slack value of the fill site column

[6]As presented, the Greedy algorithm will tend to insert fill close to the active line with minimum resistance. This may lead to worsening of critical path delay and hence cycle time in some pathological cases, compared to random fill insertion. This can be circumvented by placing an upper bound on the added net delay.

updated accordingly. These steps are repeated until fill requirements for all tiles in the layout are met.

To prevent the greedy method from quickly reaching a local minimum, we introduce two variables that enable iterations between STA and area fill synthesis.

- LB_{slack} gives a lower bound on the slack value of fill site columns. Once the largest slack value of any fill site column is less than LB_{slack}, the filling loop is stopped and a new iteration between STA and area fill synthesis is initiated with smaller LB_{slack}.
- UB_{delay} gives an upper bound on the total added delay in the layout. Once the newly added delay during an iteration exceeds UB_{delay}, the filling loop is stopped and a new iteration between STA and area fill synthesis is initiated.[7]

Our algorithm is described in detail in Figure VII.7, where the following definitions are used.

- $RF \equiv$ total number of required fill features in the given layout.
- $RF_{ij} \equiv$ number of required fill features for tile T_{ij}.
- $D_{add} \equiv$ total added delay during the current iteration.
- $S_k \equiv$ slack value of the fill site column k, which is the minimum slack value of its neighboring active lines.
- $S_{max} \equiv$ maximum slack value over all fill site columns.
- $SF_k \equiv$ maximum number of fill features that can be inserted in fill site column k such that the post-fill slack value of the column is still larger than LB_{slack}.
- $C_{k,ij} \equiv$ overlapping size of column k in tile T_{ij}.
- $F_{k,ij} \equiv$ number of inserted fill features in column k in tile T_{ij}.

[7]Since we do not budget slacks between nets, LB_{slack} and UB_{delay} serve as our means to avoid timing violations. Smaller value of LB_{slack} and larger value of UB_{delay} leads to a more accurate timing updating with the overhead of extra STA runs.

VII.G Computational Experience

We tested our proposed algorithms using three layout test cases, denoted T1, T2 and T3, obtained from industry sources. Each of the test cases was obtained in LEF/DEF format. The number of the features in T1, T2 and T3 are 24075, 22607, and 25794 respectively. Signal delay calculation is based on extracted Reduced Standard Parasitic Format (RSPF) files, and "Normal" fill was synthesized using the normal fill method [126] according to the parameters shown in the leftmost column of Table 1[8]. For these test cases with $0.38\mu m$ wire width and $0.9\mu m$ wire pitch, our design rule for the fill features has $0.9\mu m$ fill feature pitch and $0.28\mu m$ gap between fill features. The density of the fill pattern is around 50% which is the common value used in industry.[9]

VII.G.1 MDFC PIL-Fill Experiments

Table VII.1 reports the total delay increase over all wire segments due to the "normal" fill method [126], and due to our three performance-impact limited fill methods. As shown in the table, all total delay increases from the PIL-Fill methods are better than the total delay increase resulting from the normal fill method [126]. Among the PIL-Fill methods, the ILP method has the smallest delay increase (e.g, up to 97% reduction in weighted total delay increase for case $T2/80/2$, compared to the normal fill result) and its runtime is reasonable. The Greedy method is not nearly as good as the ILP method. Our experiments also show that the improvement in total delay impact depends on dissection size. As explained above, when the dissection becomes too fine grained, it becomes harder to consider the total impact of a fill site column since we handle the overlapping tiles separately.

Table VII.1: Weighted MDFC PIL-Fill synthesis. **Notation:** *T/W/r:* testcase / window size / r dissection; *Normal:* normal fill result; *ILP:* Look-up Table Based Integer Linear Programming method; *Greedy:* Greedy method; τ*:* total delay increase (ns); *CPU:* runtime of PIL-Fill step (seconds).

Testcase	**Normal**	**Greedy**		**ILP**	
T/W/r/	τ	τ	CPU	τ	CPU
T1/40/2	2583800	384942	42.9	94763.1	244.7
T1/80/2	1024280	73900.5	33.1	25758.1	320.9
T2/40/2	1264030	204194	44.2	40478.6	290.8
T2/80/2	1232160	175388	45.7	30317.4	494.1
T3/40/2	1584930	257183	40.7	69423.4	224.5
T3/80/2	587674	77341.8	34	25438.7	262.1

Table VII.2: Iterated approaches for MSFC PIL-Fill. **Notation:** *MaxDen:* maximum window density on layout; *MinDen:* minimum window density on layout; *DenConstr:* density requirement specified as a minimum post-fill window density; *MSFC-PIL:* results of MSFC PIL-Fill method; *minSlack:* minimum slack over all nets (ps).

Testcase	**Orig Layout**			**DenConstr**	**Normal**	**MSFC-PIL**	
T/W/r/	MaxDen	MinDen	minSlack	MinDen	minSlack	minSlack	CPU
T1/40/2	0.382	0.086	599.39	0.218	-202.36	454.94	273.7
T1/80/2	0.350	0.088	599.39	0.187	-215.85	473.46	369.1
T2/40/2	0.341	0.033	1061.56	0.266	400.98	896.04	357.3
T2/80/2	0.325	0.101	1061.56	0.289	308.69	883.50	571.9
T3/40/2	0.381	0.091	1974.78	0.222	1064.42	1848.18	307.8
T3/80/2	0.357	0.092	1974.78	0.191	679.24	1848.24	358.8

VII.G.2 MSFC PIL-Fill Experiments

In Table VII.2, we compare the minimum slack of all nets after the "normal" fill method and after our performance-impact limited fill method, where the density

[8] Our experimental testbed integrates GDSII Stream and internally-developed geometric processing engines, coded in C++ under Solaris 2.8. We use CPLEX version 7.0 as the integer linear programming solver. All runtimes are CPU seconds on a 300 MHz Sun Ultra-10 with 1 GB of RAM.

[9] 'Typical" sizes of dummy features depend on foundries and process nodes, as well as individual design methodologies. "Recommended" fill shapes at the 130 nm node that we are aware of range from $3.0\mu m \times 3.0\mu m$ squares to $0.6\mu m$-wide rectangles. The pitch and dimension can be varied to achieve different density targets, different mixes of area and perimeter density, or to conform to oddly-shaped available regions for fill insertion.

requirement is specified as a post-fill minimum window density. Our experiments show that the fill results from the "normal" fill method may be unacceptable with respect to the minimum slack of nets since these slack values become negative. In contrast, our iterated greedy method for MSFC PIL-Fill performs much better and all post-fill minimum slack values are still much larger than 0. The differences between the minimum slack values of "normal" fill result and MSFC PIL-Fill result show substantial advantages of our approach.

VII.H Conclusions and Future Research

In this chapter, we have developed approximations for the capacitance impact of area fill insertion, and given the first formulations for the Performance Impact Limited Fill (PIL-Fill) problem. We present two Integer Linear Programming based approaches and a Greedy method for the MDFC PIL-Fill problem, as well as an iterated greedy method for the MSFC PIL-Fill problem. Experiments on industry layouts indicate that our PIL-Fill methods can reduce the total delay impact of fill, or the impact on minimum slack, by very significant percentages.

Our ongoing research is focused on budgeting slacks along segments so that computationally expensive iteration with STA can be avoided in the optimization procedure. We are also considering slew-aware formulation of the fill insertion problem. Though Elmore delay has a high positive correlation with slew rate, an explicit slew constraint may be useful in some physical design scenarios. Other research addresses alternative PIL-Fill formulations, e.g., wherein an upper bound on timing impact constrains the minimization of layout density variation.

VII.I Acknowledgments

Chapter VII is in part a reprint of "Performance-Impact Limited Area Fill Synthesis", *ACM/IEEE Design Automation Conference*, 2003. I would like to thank my coauthors Dr. Yu Chen and Dr. Andrew B. Kahng.

Greedy MDFC PIL-Fill Algorithm

Input: the design-rule correct layout; window size w; dissection value r;
the fill pattern (size of fill feature s, gap between fill features g,
and buffer distance from interconnect b)

Output: filled layout minimizing total delay increase while satisfying density requirements

1. Partition the layout into tiles and sites
2. Run LP/Monte-Carlo [126] to get the number of required fill features (RF_{ij}) for each tile T_{ij}
3. **For** each net N_i in the layout **Do**
4. Find its intersection with each tile T_{ij} and signal direction in T_{ij}
5. Calculate entry resistances $R_l(p,q)$ of N_i in its intersected tiles
6. Run scan-line algorithm to get fill site columns in layout
7. **For** each tile T_{ij} **Do**
8. **For** each fill site column k **Do**
9. Find overlapping area of column k in tile T_{ij}
10. Get cumulative resistance $\hat{r}_k$ at position k on neighboring active lines l and l' as: $W_l(R_l(i,j) + \sum_{s=p}^{k} r_l) + W_l'(R_l'(i,j) + \sum_{s=p'}^{k} r_{l'})$
11. Calculate induced coupling capacitances $\hat{Cap}_k$ of column k as in Equation (VII.4) with C_k dummy features
12. Sort all fill columns in the tile according to their corresponding delay increments as $\hat{r}_k \cdot \hat{Cap}_k$
13. Initialize the number of filled features for tile T_{ij}: $FF_{ij} = 0$
14. **While** $FF_{ij} < RF_{ij}$ **Do**
15. Select fill column C_k with the minimum corresponding delay
16. Insert $\min((RF_{ij} - FF_{ij}), C_k)$ dummy features in the fill column
17. Delete the fill column from tile T_{ij}
18. $FF_{ij} += \min((RF_{ij} - FF_{ij}), C_k)$

Figure VII.6: Greedy MDFC PIL-Fill algorithm.

Greedy MSFC PIL-Fill Algorithm

Input: the design-rule correct layout; window size w; dissection value r;
the fill pattern (size of fill feature s, gap between fill features g,
buffer distance from interconnect b), slack lower bound LB_{slack},
and upper bound of per-iteration incremental delay UB_{delay}

Output: filled layout maximizing the minimum slack of all nets while satisfying density requirements

1. Partition the layout into tiles and sites
2. Run LP/Monte-Carlo [126] to get the number of required fill features (RF_{ij}) for each tile T_{ij} and total number of required fill features RF
3. Get fill site columns by scanning the layout
4. Run STA tool with RSPF file to get slacks for all input pins
5. Calculate the slack of each active line l
6. Calculate slack value S_k for each fill site column
7. Sort all fill site columns according to their slack values
8. **While** ($RF > 0$) **Do**
9. Choose the fill site column k with the maximum slack value S_{max}
10. **If** ($S_{max} < LB_{slack}$)
11. Update RSPF file with the capacitance increase
12. Decrease LB_{slack} by given value, **Goto** step (4)
13. Calculate SF_k for column k
14. **For** each tile T_{ij} intersecting with column k **Do**
15. Calculate overlapping size $C_{k,ij}$ of column k in tile T_{ij}
16. Number of fill features to be inserted: $F_{k,ij} = \min(RF_{ij}, C_{k,ij}, SF_k)$
17. if ($F_{k,ij} > 0$)
18. Fill up column k with $F_{k,ij}$ fill features
19. Calculate the added delay d due to $F_{k,ij}$ fill features and update neighboring active lines' delay
20. RF_{ij} -= $F_{k,ij}$, RF -= $F_{k,ij}$, SF_k -= $F_{k,ij}$, D_{add} += d;
21. if ($D_{add} > UB_{delay}$)
22. $D_{add} = 0$
23. Update RSPF file with the capacitance increase
24. **Goto** step (4)
25. Update RSPF file with the capacitance increase
26. Run STA with RSPF file to check the result

Figure VII.7: Greedy MSFC PIL-Fill algorithm.

Bibliography

[1] Cadence Encounter User's Guide, Encounter Text Command Reference.

[2] International Technology Roadmap for Semiconductors, 2003, http://public.itrs.net

[3] L. Capodieci, P. Gupta, A. B. Kahng, D. Sylvester and J. Yang, "Toward a Methodology for Manufacturability Driven Design Rule Exploration", *Proc. ACM/IEEE Design Automation Conference*, June 2004, pp. 311-316.

[4] Yuri Granik, "Correction for Etch Proximity: New Models and Applications", *Proc. SPIE*, vol. 4346, 2001, pp. 98-112

[5] A. B. Kahng and S. Mantik, "Measurement of Inherent Noise in EDA Tools", *Proc. IEEE International Symposium on Quality Electronic Design*, 2002, pp. 206-211.

[6] P. Gupta, A. B. Kahng and C.-H. Park, "Detailed Placement for Improved Depth of Focus and CD Control", *Proc. ACM/IEEE Asia South-Pacific Design Automation Conference*, Jan. 2005, pp. 343-348.

[7] P. Gupta, A. B. Kahng and C.-H. Park, "Manufacturing-aware Design Methodology for Assist Feature Correctness", *Proc. SPIE* , vol. 5756, Feb. 2005, pp. 131-140.

[8] C. S.Shi, A. K. Wong and T.-S. Ng, "Forbidden-Area Avoidance with Spacing Technique for Layout Optimization", *Proc. SPIE Design and Process Integration for Microelectronic*, 2004, Volume 5379, pp. 67-75.

[9] J. Mitra, P. Yu and D. Z. Pan, "RADAR: RET-Aware Detailed Routing Using Fast Lithography Simulations", *Proc. ACM/IEEE Design Automation Conference*, 2005, pp.369-372.

[10] P. Gupta, A. B. Kahng and C.-H. Park,"Enhanced resist and etch CD control by design perturbation", *Proc. 25th BACUS Symposium on Photomask Technology and Management*, Nov. 2005, Vol. 5992, pp. 1085-1095.

[11] K. Hashimoto, T. Kuji, Sh. Tokutome, T. Kotani, S. Tanaka and S. Inoue, "A Tandem Process Proximity Correction Method", *Proc. SPIE*, vol. 4691, 2002, pp. 1070-1081.

[12] K. Kim, Y. Choi, R. Socha and D. Flagello, "Optimization of Process Condition to Balance MEF and OPC for Alternating PSM", *Proc. SPIE*, Vol. 4691, 2002, pp. 240-246.

[13] H. Kim, D. Nam, C. Hwang, Y. Kang, S. Woo, H. Cho and W. Han, "Layer Specific Illumination Optimization by Monte Carlo Method", *Proc. SPIE*, Vol. 5040, 2003, pp. 244-250.

[14] L.W. Liebmann, "Layout Impact of Resolution Enhancement Techniques: Impediment or Opportunity?", *Proc. Proc. ACM/IEEE International Symposium on Physical Design*, 2003, pp. 110-117.

[15] M. Levenson, N. Viswanathan and R. Sympson, "Improving Resolution in Photolithography with a Phase-Shifting Mask", *IEEE Transactions on Electronic Devices*, (29), 1982, pp. 1828-1836.

[16] C.-H. Park, Y.-H. Kim, J.-S. Park, K. Kim, M.-H. Yoo and J.-T. Kong, "A Systematic Approach to Correct Critical Patterns Induced by the Lithography Process at the Full-Chip Level", *Proc. SPIE*, vol. 3679, 1999, pp. 622-629.

[17] J. Petersen, "Analytical Description of Anti-scattering and Scattering Bar Assist Features", *Proc. SPIE*, Vol. 4000, 2000, pp. 77-89.

[18] F. M. Schellenberg, L. Capodieci and R. Socha, "Adoption of OPC and the Impact on Design and Layout", *Proc. ACM/IEEE Design Automation Conference*, 2001, pp. 89-92.

[19] F. M. Schellenberg and L. Capodieci, "Impact of RET on Physical Layouts", *Proc. ACM/IEEE International Symposium on Physical Design*, 2001, pp. 52-55.

[20] X. Shi, S. Hsu, F. Chen, M. Hsu, R. Socha and M. Dusa, "Understanding the Forbidden Pitch Phenomenon and Assist Feature Placement", *Proc. SPIE*, Vol. 4562, 2002, pp. 968-976.

[21] R. Socha, M. Dusa, L. Capodieci, J. Finders, F. Chen, D. Flagello and K. Cummings, "Forbidden Pitches for 130 nm Lithography and Below", *Proc. SPIE*, Vol. 4000, 2000, pp. 1140-1155.

[22] J. Stirniman and M. Rieger, "Fast Proximity Correction with Zone Sampling", *Proc. SPIE*, vol. 2197, 1994, pp. 294-301.

[23] A. Wong, R. Ferguson, S. Mansfield, A. Molless, D. Samuels, R. Schuster and A. Thomas, "Level-Specific Lithography Optimization for 1-Gb DRAM", *IEEE Transactions on Semiconductor Manufacturing*, 13(1), 2000, pp. 76-87.

[24] J. Word, S. Zhu and J. Sturtevant, "Assist Feature OPC Implementation for the l3Onm Technology Node with KrF and No Forbidden Pitches", *Proc. SPIE*, Vol. 4691, 2002, pp. 1139-1147.

[25] A. B. Kahng, I. Markov and S. Reda, "On Legalization of Row-Based Placements", *Proc. IEEE Great Lakes VLSI Symposium*, 2004, pp. 214-219.

[26] Y. Cao, P. Gupta, D. Sylvester and J. Yang, "Design Sensitivities to Variability: Extrapolations and Assessments in Nanometer VLSI", *Proc. IEEE ASIC/SOC*, 2002, pp. 411-415.

[27] S. R. Nassif, "Design for Variability in DSM Technologies", *Proc. IEEE International Symposium on Quality Electronic Design*, 2000, pp. 451-454.

[28] S. R. Nassif, "Within-Chip Variability Analysis", *Proc. International Electronic Devices Meeting*, 1998, pp. 283-286.

[29] M. Orshansky, L. Milor, P. Chen, K. Keutzer, C. Hu, "Impact of Systematic Spatial Intra-Chip Gate Length Variability on Performance of High-Speed Digital Circuits", *Proc. ACM/IEEE International Conference on Computer Aided Design*, 2000, pp. 62-67.

[30] P. Gupta and A. B. Kahng, "Manufacturing-Aware Physical Design", *Proc. ACM/IEEE International Conference on Computer Aided Design*, 2003, pp. 681–687.

[31] L. Liebmann, G. Northrop, J. Culp, L. Sigal, A. Barish, C. Fonseca, "Layout Optimization at the Pinnacle of Optical Lithography", *Proc. SPIE*, 2003, vol. 5042, pp. 1-14.

[32] L. Liebmann, D. Maynard, K. McCullen, N. Seong, E. Buturla, M. Lavin, J. Hibbeler, "Integrating DfM Components Into a Cohesive Design-To-Silicon Solution", *Proc. SPIE*, 2005, vol. 5756, pp. 1-12.

[33] Y. Trouiller, J. Serrand, C. Miramond, F. Y. Rody, S. Manakli; P.-J. Goirand, "ArF Imaging with Off-axis Illumination and Subresolution Assist Bars: a Compromise between Mask Constraints and Lithographic Process Constraints", *Proc. SPIE*, 2002, vol. 4691, pp. 1522-1529.

[34] A. J. Lori, T. R. Michael, D. Jason, and J. Christiane, "Effect of Scattering Bar Assist Features in 193-nm Lithography", *Proc. SPIE*, 2002, vol. 4691, pp. 861-870.

[35] Design Compiler version V-2003.12, http://www.synopsys.com.

[36] A. B. Kahng and Y. C. Pati, "Subwavelength Lithography and its Potential Impact on Design and EDA", *Proc. ACM/IEEE Design Automation Conference*, 1999, pp. 799-804.

[37] L. W. Liebmann, S. M. Mansfield, A. K. Wong, M. A. Lavin, W. C. Leipold, T. G. Dunham, "TCAD Development for Lithography Resolution Enhancement", *IBM J. Res. & Dev.*, 2001, vol. 45, no. 5, pp. 651-665.

[38] S. Sirichotiyakul, T. Edwards, O. Chanhee, Z. Jingyan, A. Dharchoudhury, R. Panda and D. Blaauw, "Stand-by Power Minimization through Simultaneous Threshold Voltage Selection and Circuit Sizing", *Proc. ACM/IEEE Design Automation Conference*, 1999, pp. 436-441.

[39] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran", *Proc. International Symposium on Circuits and Systems*, May 1989, pp. 695-698.

[40] C.-P. Chen, C. C. N. Chu, and D. F. Wong, "Fast and Exact Simultaneous Gate and Wire Sizing by Lagrangian Relaxation", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 18(7), July 1999, pp. 1014-1025.

[41] Prolific, Inc. http://www.prolificinc.com/

[42] Cadence Design Systems, http://www.cadence.com/

[43] P. Gupta, Y. Kim, A. B. Kahng, D. Sylvester, "Self-Compensating Design for Focus Variation", *Proc. ACM/IEEE Design Automation Conference*, 2005, pp. 365-368.

[44] M. Orshansky, L. Milor, C. Hu, "Characterization of Spatial Intrafield Gate CD Variability, Its Impact on Circuit Performance, and Spatial Mask-Level Correction", *IEEE Transactions on Semiconductor Manufacturing*, 2004, 17(1), pp. 2-11.

[45] Y. Ye, S. Borkar, V. De, "A New Technique for Standby Leakage Reduction in High-performance Circuits", *IEEE Symposium on VLSI Circuits,* 1998, pp. 40-41.

[46] Synopsys Corp., http://www.synopsys.com/

[47] C. Visweswariah, "Death, Taxes and Failing Chips", *Proc. ACM/IEEE Design Automation Conference*, 2003, pp. 343-347.

[48] International Technology Raodmap for Semiconductors (ITRS), 2001, http://public.itrs.net.

[49] M. Orshansky and K. Keutzer, "A General Probabilistic Framework for Worst Case Timing Analysis", *Proc. ACM/IEEE Design Automation Conference*, 2002, pp. 556-561.

[50] J.A.G. Hess, K. Kalafala, S.R. Naidu, R.H.J.M. Otten, and C. Visweswariah, "Statistical Timing for Parametric Yield Prediction of Digital Integrated Circuits", *Proc. ACM/IEEE Design Automation Conference*, 2003, pp. 932-937.

[51] D. Blaauw, S. Nassif, L. Scheffer and A. Strojwas, "Design for Manufacturing in the Sub-100 nm Era", *Proc. ACM/IEEE Design Automation Conference*, Tutorial, 2003.

[52] A. B. Agrawal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Statistical Timing Analysis Using Bounds and Selective Enumeration", *Proc. ACM/IEEE Design Automation Conference*, 2003, pp. 348-353.

[53] William Chu, IBM Corp., *Personal Communication*, July 2003.

[54] S. Postnikov and S. Hector, "ITRS CD Error Budgets: Proposed Simulation Study Methodology", May 2003.

[55] R.A. Budd, D.B. Dove, J.L. Staples, R.M. Martino, R.A. Ferguson, and J.T. Weed, "Development and Application of a New Tool for Lithographic Mask Evaluation, the Stepper Equivalent Aerial Image Measurement System, AIMS", *IBM Journal of R&D*, 1997, 41(12), pp. 119-129.

[56] ASML MaskTools Inc., http://www.masktools.com/content/scat_bars.pdf

[57] Prolith version 8.0, *http://www.kla-tencor.com*

[58] A. Agarwal, C. H. Kim, S. Mukhopadhyay and K. Roy, "Leakage in Nano-Scale Technologies: Mechanisms, Impact and Design Considerations", *Proc. ACM/IEEE Design Automation Conference*, 2004, pp. 6-11.

[59] F. Beeftink, P. Kudva, D. Kung and L. Stok, "Combinatorial Cell Design for CMOS Libraries", *Integration, the VLSI Journal*, 2000, 29(4), pp. 67-93.

[60] F. Boeuf et al., "A Conventional 45nm CMOS Node Low-Cost Platform for General Purpose and Low-Power Applications", *IEEE International Electron Devices Meeting*, 2004, pp. 425-428.

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

[62] J. Halter and F. Najm, "A Gate-level Leakage Power Reduction Method for Ultra low-power CMOS Circuits", *IEEE Custom Integrated Circuits Conference*, 1997, pp. 475-478.

[63] M. Horiguchi, T. Sakata and K. Itoh, "Switched-Source-Impedance CMOS Circuit for Low Standby Sub-Threshold Current Giga-Scale LSI's", *IEEE Journal of Solid-State Circuits*, 1993, 28(11), pp. 1131-1135.

[64] I. Hyunsik, T. Inukai, H. Gomyo, T. Hiramoto and T. Sakurai, "VTCMOS Characteristics and its Optimum Conditions Predicted by a Compact Analytical Model", *Proc. ACM/IEEE International Symposium on low-power Electronics and Design*, 2001, pp. 123-128.

[65] J. Kao, S. Narendra and A. Chandrakasan, "MTCMOS Hierarchical Sizing Based on Mutual Exclusive Discharge Patterns", *Proc. ACM/IEEE Design Automation Conference*, 1998, pp. 495-500.

[66] M. Ketkar and S. Saptnekar, "Standby Power Optimization via Transistor Sizing and Dual Threshold Voltage Assignment", in *Proc. ACM/IEEE International Conference on Computer Aided Design*, 2002, pp. 375-378.

[67] D. Lee and D. Blaauw, "Static Leakage Reduction Through Simultaneous Threshold Voltage and State Assignment", *Proc. ACM/IEEE Design Automation Conference*, 2003, pp. 192-194.

[68] Z. Luo, "High Performance and low-power Transistors Integrated in 65 nm Bulk CMOS Technology", *IEEE International Electron Devices Meeting*, 2004, pp. 661-664.

[69] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu and J. Yamada, "1-V Power Supply High-Speed Digital Circuit Technology with Multithreshold-Voltage CMOS", *IEEE Journal of Solid-State Circuits*, 1995, 30(8), pp. 847-854.

[70] S. Mutoh, S. Shigematsu, Y. Matsuya, H. Fukada, T. Kaneko and J. Yamada, "1V Multithreshold-Voltage CMOS Digital Signal Processor for Mobile Phone Applications", *IEEE Journal of Solid-State Circuits*, 1996, 31(11), pp. 1795-1802.

[71] Y. Nakahara et al., "A Robust 65-nm Node CMOS Technology for Wide-Range Vdd Operation", *IEEE International Electron Devices Meeting*, 2003, pp. 11.2.1-11.2.4.

[72] S. Nakai et al., "A 65 nm CMOS Technology with a High-Performance and Low-Leakage Transistor, a 0.55 /spl mu/m/sup 2/ 6T-SRAM Cell and Robust Hybrid-ULK/Cu Interconnects for Mobile Multimedia Applications", *IEEE International Electron Devices Meeting*, 2003, pp. 11.3.1-11.3.4.

[73] S. Narendra, D. Blaauw, A. Devgan and F. Najm, "Leakage Issues in IC Design: Trends, Estimation and Avoidance", *Proc. ACM/IEEE International Conference on Computer Aided Design*, 2003, tutorial.

[74] K. Nose, M. Hirabayashi, H. Kawaguchi, S. Lee and T. Sakurai, "V_{th} Hopping Scheme to Reduce Subthreshold Leakage for Low-Power Processors," *IEEE Journal of Solid-State Circuits*, 37(3), 2002, pp. 413-419.

[75] R. Rao, A. Srivastava, D. Blaauw and D. Sylvester, "Statistical analysis of subthreshold leakage current for VLSI circuits", *IEEE Transactions on Very Large Scale Integrated Systems*, 12(2), 2004, pp. 131-139.

[76] P. Royannez et al., "90 nm Low Leakage SoC Design Techniques for Wireless Applications", *IEEE International Solid-State Circuits Conference*, 2005, pp. 138-589.

[77] D. K. Schroder and J. A. Babcock, "Negative Bias Temperature Instability: Road to Cross in Deep Submicron Silicon Semiconductor Manufacturing", *Journal of Applied Physics*, 2003, 94(1), pp. 1-18.

[78] S. Shigematsu, S. Mutoh, Y. Matsuya, Y. Tabae and J. Yamada, "A 1-V High-Speed MTCMOS Circuit Scheme for Power-Down Application Circuits", *IEEE Journal of Solid-State Circuits*, 1997, 32(6), pp. 861-869.

[79] S. Sirichotiyakul, T. Edwards, C. Oh, R. Panda and D. Blaauw, "Duet: An Accurate Leakage Estimation and Optimization Tool for Dual-V_{th} Circuits", *IEEE Transactions on Very Large Scale Integrated Systems*, 2002, vol. 10, pp. 79-90.

[80] S. Sirichotiyakul, T. Edwards, C. Oh, J. Zuo, A. Dharchoudhury, R. Panda and D. Blaauw, "Stand-by Power Minimization through Simultaneous Threshold Voltage Selection and Circuit Sizing", *Proc. ACM/IEEE Design Automation Conference*, 1999, pp. 436-441.

[81] N. Sirisantana, L. Wei and K. Roy, "High-Performance Low-Power CMOS Circuits Using Multiple Channel Length and Multiple Oxide Thickness", *Proc. IEEE International Conference on Computer Design*, 2000, pp. 227-232.

[82] L. Wei, Z. C. andn M. Johnson, K. Roy and V. De, "Design and Optimization of Low Voltage High Performance Dual Threshold CMOS Circuits", *Proc. ACM/IEEE Design Automation Conference*, 1998, pp. 489-494.

[83] L. Wei, Z. Chen and K. Roy, "Mixed-V_{th} CMOS Circuit Design Methodology for low-power Applications", *Proc. ACM/IEEE Design Automation Conference*, 1999, pp. 430-435.

[84] L. Wei, K. Roy and C. K. Koh, "Power Minimization by Simultaneous Dual-V_{th} Assignment and Gate-Sizing", *Proc. IEEE Custom Integrated Circuits Conference*, 2000, pp. 413-416.

[85] F. Brglez, D. Bryan, and K. Kozminski, "Combinatorial Profiles of Sequential Benchmark Circuits," *Proc. International Symposium on Circuits and Systems* , 1989, pp. 1929-1934.

[86] F.-L. Heng, P. Gupta, K. Lai, R. L. Gordon, J.-F. Lee, "Taming Pattern and Focus Variation in VLSI Design", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2004, pp. 139-148.

[87] G. Czech, E. Richter and O. Wunnicke, "193nm Resists: A Status Report", *Future Fab Intl. Volume 12*, 2002.

[88] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker and S. Narayan,"First Order Incremental Block-Based Statistical Timing Analysis", *Proc. ACM/IEEE Design Automation Conference*, 2004, pp. 331-336.

[89] A. P. Balasinski, L. Karklin, and V. Axelrad, "Impact of Subwavelength CD Tolerance on Device Performance", *Proc. SPIE Conference on Design, Process Integration, and Characterization for Microelectronics*, Vol. 4692, Jul. 2002, pp. 361-368.

[90] R. C. Pack, V. Axelrad, A. Shibkov, V. Boksha, J. A. Huckabay, R. Salik, W. Staud, R. Wang, W. D. Grobman, "Physical and Timing Verification of Subwavelength-Scale Designs: I. Lithography Impact on MOSFETs", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, Vol. 5042, Jul 2003, pp. 51-62.

[91] F.-L. Heng, J.-F. Lee, P. Gupta, "Toward Through-Process Layout Quality Metrics", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing III*, Vol. 5756, May 2005, pp. 161-167.

[92] S.D. Kim, H. Wada and Jason C.S. Woo, "TCAD-Based Statistical Analysis and Modeling of Gate Line-Edge Roughness Effect on Nanoscale MOS Transistor Performance and Scaling", *IEEE Transactions on Semiconductor Manufacturing*, 17(2), May 2004, pp. 192-200.

[93] L.A. Akers "The Inverse-Narrow-Width Effect", *IEEE Electron Device Letters*, EDL-7(7), July 1986, pp. 419-421.

[94] C. Pacha, M. Bach, K. V. Arnim, R. Brederlow, D. S. Lansiedel, P. Seegebrecht, J. Berthold, and R. Thewes, "Impact of STI-Induced Stress, Inverse Narrow Width Effect, and Statistical V_{th} Variations on Leakage Current in 120nm CMOS", *Proc. European Solid-State Device Research conference*, 2004, pp. 397- 400.

[95] P. Oldiges, Q. Lin, K. Petrillo, M. Sanchez, M. Ieong and M. Hargrove, "Modeling Line Edge Roughness Effects in sub 100 Nanometer Gate Length Devices", *Proc. International Conference on Simulation of Semiconductor Processes and Devices*, 2000, pp.131-134.

[96] I. Polishchuk, N. Mathur, C. Sandstrom, P. Manos and O. Pohland, "CMOS Vt-control Improvement through Implant Lateral Scatter Elimination", *IEEE International Symposium on Semiconductor Manufacturing*, 2005, pp. 193-196

[97] *BSIM4.5.0 User's Manual*, 2005

[98] K. K.-L. Hsueh, J. J. Sanchez, T. A. Demassa and L. A. Akers, "Inverse-Narrow-Width Effects and Small-Geometry MOSFET Threshold Voltage Model", *IEEE transactions on Electron Devices*, 35(3), March 1988, pp. 325-338

[99] *Davinci User's Guide*, Version W-2004.09, Sep. 2004

[100] *Mathematica 5.2,* http://www.wolfram.com/

[101] N. B. Cobb and Y. Granik, "Using OPC to Optimize for Image Slope and Improve Process Window", *Proc. of SPIE*, Vol. 5130, 2003, pp. 838-843.

[102] D.G. Flagello, H. van der Laan, J. van Schoot, I. Bouchoms and B. Geh, "Understanding Systematic and Random CD Variations Using Predictive Modeling Techniques", *SPIE Conference Optical Microlithography XII*, 1999, pp. 162-175.

[103] A. Goda, A. Mikasa, S. Odanaka, S. Kobayashi and H. Watanabe, "Improvements of RSF for Statistical Design of Lithographic Process", *International Workshop on Statistical Metrology*, 1997, pp. 74-77.

[104] P. Gupta and F.-L. Heng, "Toward a Systematic-Variation Aware Timing Methodology", *Proc. ACM/IEEE Design Automation Conference*, 2004, pp. 321-326.

[105] A. B. Kahng, S. Muddu, P. Sharma, "Defocus-Aware Leakage Estimation and Control", *Proc. ACM/IEEE International Symposium on low-power Electronics and Design*, 2005, pp. 263-268.

[106] L. W. Liebmann, A. F. Molless, R. A. Ferguson, A. K. K. Wong, S. M. Mansfield, "Understanding Across-Chip Line-Width Variation: The First Step Toward Optical Proximity Correction", *Proc. of SPIE*, Vol. 3051, 1997, pp. 124-136.

[107] A. Mikasa, A. Goda, K. Matsuoko, H. Umimoto and S. Odanaka, "A Statistical Critical Dimension Control at CMOS cell-level", *International Electron Devices Meeting*, 1996, pp. 631-634

[108] A. Mikasa, A. Goda, S. Odanaka, S. Kobayashi and H. Watanabe, "A Statistical Gate CD Control Including OPC", *Symposium on VLSI Technology*, 1998, pp. 170-171.

[109] T. Roessler and J. Thiele, "Geometrical Analysis of Product Layout as a Powerful Tool for DFM", *Proc. of SPIE*, Vol. 5756, 2005, pp. 150-160.

[110] A. K. K. Wong, A. F. Molless, T. A. Brunner, E. Coker, R. H. Fair, G. L. Mack, S. M. Mansfield, "Linewidth Variation Characterization by Spatial Decomposition", *Journal of Microlithography, Microfabrication and Microsystems I*, 2002(1), pp. 106-116.

[111] J. Yang, L. Capodieci, and D. Sylvester, "Advanced Timing Analysis Based on Post-OPC Extraction of Critical Dimensions", *Proc. ACM/IEEE Design Automation Conference*, 2005, pp. 359-364.

[112] P. Gupta, A. B. Kahng, S. Nakagawa, S. Shah and P. Sharma, "Lithography Simulation-Based Full-Chip Design Analyses", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, Volume 6156, 2006, pp. 277-284.

[113] International Technology Roadmap for Semiconductors: Yield Enhancement, 2005, *http://public.itrs.net*

[114] International Technology Roadmap for Semiconductors: Front-End Processes, 2005, *http://public.itrs.net*

[115] P. Gupta, A. B. Kahng, C.-H. Park, K. Samadi and X. Xu, "Wafer Topography-Aware Optical Proximity Correction", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 25, 2006, pp. 2749-2756.

[116] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi and V. De, "Parameter Variations and Impact on Circuits and Microarchitecture", *Proc. ACM/IEEE Design Automation Conference*, 2003, pp. 338-342

[117] A. Salz and M. Horowitz, "IRSIM: an Incremental MOS Switch-Level Simulator", *Proc. ACM/IEEE Design Automation Conference*, 1989, pp. 173-178

[118] "International Technology Roadmap for Semiconductors," http://public.itrs.net

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

[120] A. B. Kahng, C.-H. Park, P. Sharma and Q. Wang, "Lens Aberration-Aware Placement for Across Field Line-Width Control", in *Proc. ACM/IEEE Design Automation and Test in Europe*, 2006, to appear.

[121] M. Orshansky, L. Milor, P. Chen, K. Keutzer and C. Hu, "Impact of Spatial Intrachip Gate Length Variability on the Performance of High-Speed Digital Circuits", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 21(5), 2002, pp. 544-553.

[122] N. D. Arora, K. V. Raol, R. Schumann, and L. M. Richardson, "Modeling and Extraction of Interconnect Capacitances for Multi-layer VLSI Circuits", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems* 15(1), 1996, pp. 58-67.

[123] E. Barke, "Line-to-Ground Capacitance Calculation for VLSI: A Comparison", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems* 7(2), 1988, pp. 295-298.

[124] R. Bek, C. C. Lin and J. H. Liu, *Personal Communication*, December 1997.

[125] D. Boning and J. Chung, "Statistical Metrology - Measurement and Modeling of Variation for Advanced Process Development and Design Rule Generation", *Int. Conference on Characterization and Metrology for ULSI Technology*, Gaithersburg, MD, March 1998, vol. 448, pp. 395-404.

[126] Y. Chen, A. B. Kahng, G. Robins and A. Zelikovsky, "Dummy Fill Synthesis for Uniform Layout Density", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems* 21(10), 2002, pp. 1132-1147.

[127] Y. Chen, A. B. Kahng, G. Robins and A. Zelikovsky, "Smoothness and Uniformity of Filled Layout for VDSM Manufacturability", *Proc. ACM/IEEE International Symposium on Physical Design*, April 2002, pp. 137-142.

[128] Y. Chen, P. Gupta and A. B. Kahng, "Performance-Impact Limited Dummy Fill Insertion", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, Feb. 2003.

[129] Y. Chen, P. Gupta and A. B. Kahng, "Performance-Impact Limited Area Fill Synthesis", *Proc. ACM/IEEE Design Automation Conference*, 2003, pp. 22-27.

[130] J. Chern, J. Huang, L. Aldredge, P. Li and P. Yang, "Multilevel Metal Capacitance Models for CAD Design Synthesis Systems", *IEEE Electron Device Letters* 13(1), 1992, pp. 32-34.

[131] J. Cong, L. He, A. B. Kahng, D. Noice, N. Shirali and S. H.-C. Yen, "Analysis and Justification of a Simple, Practical 2 1/2-D Capacitance Extraction Methodology", *Proc. ACM/IEEE Design Automation Conference*, 1997, pp. 627-632.

[132] W. C. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers", *Journal of Applied Physics*, vol. 1, 1948, pp. 55-63.

[133] W. Grobman, M. Thompson, R. Wang, C. Yuan, R. Tian and E. Demircan, "Reticle Enhancement Technology: Implications and Challenges for Physical Design ", *Proc. ACM/IEEE Design Automation Conference*, 2001, pp. 73-78.

[134] A. B. Kahng, S. Muddu and E. Sarto, "On Switch Factor Based Analysis of Coupled RC Interconnects", *Proc. ACM/IEEE Design Automation Conference*, 2000, pp. 79-84.

[135] H. Landis, P. Burke, W. Cote, W. Hill, C. Hoffman, C. Kaanta, C. Koburger, W. Lange, M. Leach and S. Luce, "Integration of Chemical-Mechanical Polishing into CMOS Integrated Circuit Manufacturing", *Thin Solid Films*, vol. 220, 1992, pp. 1-7.

[136] W. S. Lee, K. H. Lee, J. K. Park, T. K. Kim, Y. K. Park and J. T. Kong, "Investigation of the Capacitance Deviation Due to Metal-Fills and the Effective Interconnect Geometry Modeling", *Proc. IEEE International Symposium on Quality Electronic Design*, 2003.

[137] G. Nanz and L. E. Camilletti, "Modeling of Chemical-Mechanical Polishing: A Review", *IEEE Transactions on Semiconductor Manufacturing*, 8(4), 1995, pp. 382-389.

[138] T. Sakurai, "Closed-Form Expressions for Interconnect Delay, Coupling, Crosstalk in VLSI's", *IEEE Transactions on Electron Devices*, vol. ED-40, 1993, pp. 118-124.

[139] *Praesagus, Inc.*, `http://www.praesagus.com/`

[140] R. Radojcic, *Personal Communication*, March 1996.

[141] B. E. Stine, D. S. Boning et al., "The Physical and Electrical Effects of Metal Fill Patterning Practices for Oxide Chemical Mechanical Polishing Processes", *IEEE Transactions on Electron Devices*, 45(3), 1998, pp. 665-679.

[142] A. Toulouse, D. Bernard, C. Landrault and P. Nouet "Efficient 3D Modeling for Extraction of Interconnect Capacitances in Deep Submicron Dense Layouts", *Proc. Design Automation and Test Europe*, 1999, pp. 576-580.

[143] *UbiTech. Inc.*, `http://www.ubitechnology.com/`

[144] *XYALIS*, `http://www.xyalis.com/`

[145] M. L. Cote, P. Hurat, A. Miloslavsky, D. Goinard and M. L. Rieger, "Mask Cost Reduction and Yield Optimization Using Design Intent", *Proc. SPIE on Design and Process Integration for Microelectronic Manufacturing*, 2005, pp. 389-396.

[146] P. Gupta, A. B. Kahng, D. Sylvester and J. Yang, "A Cost-Driven Lithographic Correction Methodology Based on Off-the-Shelf Sizing Tools", *Proc. ACM/IEEE Design Automation Conference*, 2003, pp. 16-21.

[147] M. E. Mason, "Rising Cost of RETs: Understanding the Value Proposition", *Proc. SPIE on Design and Process Integration for Microelectronic Manufacturing*, 2004, pp.10-19.

[148] M. L. Rieger, *Mask EDA Workshop*, 2001, `http://www.sematech.org/resources/litho/meetings/mask/20010711/G_AVANTI.pdf` .

[149] S. F. Schulze and J. Word, "Interaction of RET and MDP: Optimization for Reducing the Mask Writing Time", *Proc. SPIE on Design and Process Integration for Microelectronic Manufacturing*, 2004, pp. 170-181.

[150] http://www.insightful.com/products/splus/

[151] Y. Zhang, S. Chou, B. Rockwell, G. Xiao, H. H. Kamberian, R. Cottle and C. J. Progler, "Mask Cost Analysis via Write-time Estimation", *Proc. SPIE on Design and Process Integration for Microelectronic Manufacturing*, vol. 5756, 2005, pp. 313-318.

[152] P. Gupta, A. B. Kahng, D. Sylvester, and J. Yang, "Performance-Driven OPC for Mask Cost Reduction", *Proc. IEEE International Symposium on Quality Electronic Design*, 2005, pp. 270-275, .

[153] C. Yang, "Challenges of Mask Cost and Cycle Time", *SEMATECH: Mask Supply Workshop*, Intel, 2001.

[154] P. Gupta, F.-L. Heng and M. Lavin, "Merits of Cellwise Model-Based OPC", *Proc. SPIE International Symposium on Microlithography*, vol. 5379, 2004, pp. 182-189.

[155] S. Murphy, Dupont Photomask, *SEMATECH: Mask Supply Workshop*, 2001.

[156] M. L. Rieger, J. P. Mayhew and S. Panchapakesan, "Layout Design Methodologies for Sub-Wavelength Manufacturing", *Proc. ACM/IEEE Design Automation Conference*, 2001, pp. 85-88.

[157] *Optical Lithography Cost of Ownership - Final Report* , http://www.sematech.org/docubse/document/4014atr.pdf

[158] K. Wampler, ASML MaskTools, Personal Communication, March 2003.

[159] R. Nair, C. L. Berman, P. S. Hauge and E. J. Yoffa, "Generation of Performance Constraints for Layout", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 8(8), 1989, pp. 860-874.

[160] C. Chen, E. Bozorgzadeh, A. Srivastava and M. Sarrafzadeh, "Budget Management with Applications", *Algorithmica*, vol. 34, 2002, pp. 261-275.

[161] E. Bozorgzadeh, S. Ghiasi, A. Takahashi and M. Sarrafzadeh, "Optimal Integer Delay Budgeting on Directed Acyclic Graphs", *Proc. ACM/IEEE Design Automation Conference*, 2003, pp. 920-925.

[162] http://www.mentor.com

[163] http://www.opencores.org

[164] http://www.ilog.com

[165] Y. Zhang, R. Gray, O. S. Nakagawa, P. Gupta, H. Kamberian, G. Xiao, R. Cottle, and C. Progler, "Interaction and Balance of Mask Write Time and Design RET Strategies", *Proc. SPIE Photomask and Next Generation Lithography Mask Technology*, Japan, vol. 5843, 2005, pp. 614-618.

[166] A. Agarwal, D. Blaauw and V. Zolotov, "Statistical Timing Analysis for Intra-Die Process Variations with Spatial Correlations", *Proc. ACM/IEEE International Conference on Computer Aided Design*, 2003, pp. 900-907.

[167] A. B. Kahng, "Research Directions for Coevolution of Rules and Routers" , *Proc. ACM/IEEE International Symposium on Physical Design*, 2003, pp. 122-125

[168] P. Rabkin, "DFM for Advanced Technology Nodes: Fabless View", *Future Fab International*, vol. 20, 2006, http://www.future-fab.com, .