PERFORMANCE-BASED
OPTICAL PROXIMITY CORRECTION

TEH SIEW HONG

(B.Eng. (Hons., 1st Class), UM)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2011

Acknowledgements

My journey of pursuing graduate studies at National University of Singapore has been a fulfilling and intellectually challenged one. With this opportunity, I wish to acknowledge and note my heartfelt appreciation of those people who stand out most notably in my mind as contributing to the production and success of this thesis.

First of all, my sincerest gratitude and thanks go to my supervisors, Assoc. Prof. Arthur Tay and Dr. Heng Chun Huat, for their support, guidance and encouragement during my graduate years in National University of Singapore. Without their consistent involvements, stimulating ideas, suggestions and help in every aspect of my research, this thesis would not have been possible.

I would also like to thank National University of Singapore for the financial support given through the Research Scholarship and the President's Graduate Fellowship, as well as for the academic support provided via intellectual and resourceful lecturers, helpful staffs and excellent library services. Special gratitude goes to Prof. Lee Tong Heng, Assoc. Prof. Ho Weng Khuen, Prof. Ben Chen, Prof. Wang Qing-Guo, Prof. Lian Yong, Dr. Yao Li Bing, Assoc. Prof. Xu Yong Ping, Dr. Heng Chun Huat, Assoc. Prof. Tan Kay Chen and Dr. Peter Chen who have taught me in class and imparted me extensive knowledge in control, circuit design and computer intelligence. Not forgetting also, many thanks to Ms. Vathi, Ms. Zheng Huan Quan, Ms Wang Ying, Mr. Lin Jian Qiang for their utmost technical and logistical support from time to time.

Next, I would like to thank all my friends and colleagues who have shared inspiring experiences and entertaining moments with me: Dr. Wang Yuheng, Dr.

Shao Lichun, Dr. Chua Teck Wee, Dr. Quek Han Yang, Dr. Lim Li Hong, Dr. Yan Han, Mr. Hong Choo Yang, Mr. Lee See Chek, Mr. Lai Chow Yin, Mr. Chew Xiong Yue, Mr. Tan Yew Teck, Ms. Do Thi Thu Trang, Mr. Tan Yen Kheng, Mr. Ngo Yit Sung, Mr. Yong See Wei, Mr. Feng Yong, Mr. Nie Maowen, Mr. Qu Yifan, Mr. Yang Geng, Ms. Sun Yajuan, Mr. Ang Kar Tien , and many others at the Advanced Control Technology Laboratory (ACT). Thanks to all whom I have unintentionally left out, but contributing in a way making this thesis a success or making my journey a memorable one.

Lastly and most importantly, I wish to express my sincere appreciation to my family for their love and support which have always been a constant source of strength for me. Especially thanks to my parents who have raised me up and showered me with their unconditional love and care. I also owe my loving thanks to my husband Kent Yeoh, who has been very supportive and his constant encouragement has actually brought me till the end of this journey. To them, I dedicate this thesis …

Teh Siew Hong

December, 2010.

Contents

Acknowledgements i

Contents iii

Summary.. vi

List of Tables viii

List of Figures.. ix

1. Introduction 1

1.1 Background...1

1.1.1 Overview of Optical Proximity Correction (OPC)......................5

1.1.2 Historical Perspectives of OPC 8

1.1.3 Challenges and Motivation...12

1.2 Contributions..14

1.2.1 Design-process Integration for Performance-based OPC (PB-OPC) Framework 14

1.2.2 Device Performance-based OPC (DPB-OPC) Methodology....15

1.2.3 Library-based Device Performance-based OPC for Hierarchical Circuits 16

1.2.4 Device Current and Capacitance Oriented OPC (IC-OPC).......18

1.3 Organization...18

2. Design-process Integration for Performance-based OPC (PB-OPC) Framework...20

2.1 Introduction...20

2.2 The Proposed PB-OPC Framework 23
2.2.1 Device Characteristic Library 24
2.2.2 Designed Performance Extraction..24
2.2.3 Lithography Process Simulation 25
2.2.4 Printed Transistor Performance Extraction 25
2.2.5 Mask Generation Algorithm...28

2.3 Results and Discussions...30
2.3.1 Transistor NMOS and PMOS...32
2.3.2 Standard Digital Cells 34
2.3.3 Six-stage Inverter Chain...35
2.3.4 4-Bytes 6T-SRAM Cell..37

2.4 Chapter Summary 40

3. Device Performance-based OPC (DPB-OPC) Methodology 41

3.1 Introduction...41

3.2 DPB-OPC Methodology 42
3.2.1 Performance Extraction Model 44
3.2.2 DPB-OPC Mask Design Algorithm 45

3.3 Results and Discussions...53
3.3.1 Performance-optimized EPE-OPC Mask Generation 54
3.3.2 Comparison with EPE-OPC Methodology...............................56
3.3.3 Investigation of Post-OPC Path Delay 63

3.4 Chapter Summary 67

4. Library-based Device Performance-based OPC for Hierarchical Circuits 68

4.1 Introduction...68

4.2 Library-based DPB-OPC Flow 70
4.2.1 Library Database 71
4.2.2 Library-based DPB-OPC Mask Generation Algorithm 72

4.3 Results and Discussions 77

4.4 Chapter Summary 81

5. Device Current and Capacitance Oriented OPC (IC-OPC) 83

5.1 Introduction 83

5.2 Overview of IC-OPC Flow 84
5.2.1 IC-OPC Mask Synthesizer 86

5.3 Results and Discussions 89
5.3.1 Post-OPC Performance Deviation 89
5.3.2 Mask Size 94
5.3.3 Run Time 94

5.4 Chapter Summary 95

6. Conclusion 96

6.1 Summary 96

6.2 Future Work 99

Author’s Publications 101

Bibliography 103

Summary

Lithography continues to be the key technology driver in today's semiconductor manufacturing. The ability of extending the existing exposure system into sub-wavelength printing regime is enabled by resolution enhancement techniques such as optical proximity correction (OPC). ITRS projects OPC getting more difficult and expensive to implement at each successive technology generation. Therefore it is of immense interest to research new techniques to reduce the cost of OPC. In this thesis, the development and analysis of circuit performance driven OPC frameworks are presented to reduce mask costs and improve circuit performance matching.

A design-process integrated performance-based OPC (PB-OPC) framework is first developed to generate simpler OPC mask that achieves closer circuit performance. It exploits the design intent extracted from the design layout to guide upon the customized OPC mask generator. The feasibility of the proposed PB-OPC framework is demonstrated via simulation results compared to a commercial OPC tool. The simulation results reveal that PB-OPC outperforms the conventional edge placement error based OPC (EPE-OPC) approach in two aspects: reduction in mask data volume and circuit performance variation over the various test cases.

A complete device performance-based OPC (DPB-OPC) framework is further generalized and presented. The non-linear current density along the channel width due to threshold voltage variation and edge effect is addressed with a weighted gate-slicing method. A systematic approach to determine the initial mask adjustment step is proposed to speed up the computation and this has resulted in additional 3.07% reduction in mean drive current (*Ion*) deviation compared to PB-OPC. In addition, a DRC compliance regulator is also developed for design rule checking to ensure that the post-OPC printed patterns are free from bridging, pinching, open or short issues. By simulation, DPB-OPC outperforms the performance-optimized EPE-OPC approach in two aspects: an average of 34% reduction in mask size and up to 13.5% reduction in device performance deviation.

Next, a library-based DPB-OPC framework is developed to handle the synthesized digital circuit. By making use of the hierarchical information of the synthesized circuit and the pre-characterized DPB-OPC library, the OPC run time efficiency is greatly improved. Simulation demonstrates that the library-based DPB-OPC approach has performance comparable to full chip DPB-OPC, but with run time reduction of up to 44× in the ISCAS'85 benchmark design.

Finally, a hybrid *Ion* and capacitance based OPC (IC-OPC) is proposed to achieve satisfactory co-matching on both *Ion* and gate capacitance in digital circuit. The performance deviation error is the weighted sum of *Ion* and gate capacitance error. The customized mask synthesizer alters the mask according to the decision matrix, which is constructed based on the relationship between *Ion*, gate capacitance with respect to channel width and length. Simulation shows that IC-OPC outperforms the performance-optimized EPE-OPC approach in three aspects: an average of 32% reduction in mean path delay deviation, an average of 34% reduction in mask size and at least 84% of run time saving.

List of Tables

Table 1.1: Various techniques for achieving desired *CD* control and overlay with optical projection lithography [12]..4

Table 2.1: The three supported operation modes in PB-OPC...........................29

Table 2.2: 65nm NMOS transistor...34

Table 2.3: 65nm PMOS transistor..34

Table 2.4: Standard digital cells...35

Table 2.5: Six-stage inverter chain...37

Table 2.6: 4-Bytes 6T-SRAM cell. 38

Table 3.1: Benchmark circuit specification. 54

Table 3.2: Comparison of post-OPC circuit performance. 62

Table 3.3: Comparison of mask size..62

Table 3.4: Comparison of OPC run time. 62

Table 3.5: Comparison of post-OPC path delay deviation. 65

Table 4.1: Comparison of post-OPC device performance error. 79

Table 4.2: Comparison of run time...80

Table 4.3: Comparison of run time for different MAXDIFF settings..............80

Table 5.1: Normalized path delay deviation with respect to EPE-OPC. 91

Table 5.2: Normalized mask size with respect to EPE-OPC............................94

Table 5.3: Normalized run time with respect to EPE-OPC search time...........95

Table 6.1: Comparison of various OPC frameworks.......................................98

List of Figures

Figure 1.1: Typical steps in the lithography sequence [5]. 2

Figure 1.2: Projection exposure system. [7] 3

Figure 1.3: OPC improves layout-to-wafer pattern fidelity. 5

Figure 1.4: Typical image fidelity problems in lithography [13]. 6

Figure 1.5: Line end shortening impacts overlay control and circuit density [9]. 7

Figure 1.6: Methods for line end shortening reduction [9]. 7

Figure 1.7: Corner rounding. 8

Figure 1.8: Simplified diagram for the forward model-based OPC flow [23-25]. 11

Figure 1.9: Mask with (a) no OPC (b) medium aggressive OPC (c) aggressive OPC scheme. 11

Figure 2.1: Flowchart of the proposed PB-OPC framework............................24

Figure 2.2: Performance extraction for nonrectangular gate. 27

Figure 2.3: Estimation of the transistor’s current characteristic. 27

Figure 2.4: Segmentation process. 29

Figure 2.5: Flowchart of the mask generation algorithm. 30

Figure 2.6: Cost function plot within the search space (step = 1nm). Locally optimal OPC setting is frag 20nm, step 1nm, iter 2. 32

Figure 2.7: Comparison between EPE-OPC mask (frag 65nm, step 2nm, iter 3) and PB-OPC mask. 33

Figure 2.8: Layout of the six-stage inverter chain. 36

Figure 2.9: Comparison between EPE-OPC mask (frag 30nm, step 1nm, iter 4) and PB- OPC mask. 37

Figure 2.10: Layout of the 4-Bytes 6T-SRAM cell. 38

Figure 2.11: Comparison between EPE-OPC mask (frag 20nm, step 1nm, iter 2) and PB-OPC mask. 39

Figure 2.12: Butterfly plots of the original design, EPE-OPC and PB-OPC....39

Figure 3.1: The proposed DPB-OPC flow. 42

Figure 3.2: Graphical illustration of segmentation process. 46

Figure 3.3: Flowchart of the DPB-OPC mask design algorithm. 47

Figure 3.4: Dependency of current *Ion* on *W* and *L*. 49

Figure 3.5: Model for characterizing *init_adjust* (dimension is in nanometer).51

Figure 3.6: Performance difference between DPB-OPC and PB-OPC [65].....51

Figure 3.7: Before and after the DRC compliance regulator. 53

Figure 3.8: Mean *Ion* deviation varies with OPCpro setting. 55

Figure 3.9: Correlation of transient performance with mean *Ion* deviation error. 58

Figure 3.10: Performance difference between DPB-OPC and EPE-OPC. 59

Figure 3.11: Mask size comparison (diffusion and polysilicon masks) between DPB-OPC and EPE-OPC..60

Figure 3.12: Run time comparison between DPB-OPC and EPE-OPC. 61

Figure 3.13: Comparison between (a) DPB-OPC mask and (b) EPE-OPC mask of circuit c432. 63

Figure 3.14: Histogram of path delay deviation for (a) post DPB-OPC c1908 and (b) post EPE-OPC c1908. 66

Figure 3.15: Histogram of gate area deviation for (a) post DPB-OPC c1908 and (b) post EPE-OPC c1908. 66

Figure 3.16: Trendline relationship between the mean gate area deviation and the designed transistor width. 66

Figure 4.1: Library-based DPB-OPC methodology. 71

Figure 4.2: Layout and cell view for synthesized circuit c432. 73

Figure 4.3: Layout in GDSII and CIF text format. 74

Figure 4.4: The differences in the post-lithography print image of gate regions (partial C432b circuit) are highlighted in blue region. 75

Figure 4.5: Distribution of Δ*Ion_error* of c432b circuit. The average Δ*Ion_error* is 0.04%. 77

Figure 4.6: Plot of gate with poorer Δ*Ion_error* and max Δ*Ion_error* for all ISCAS85' test cases. 77

Figure 4.7: Reduction of *Ion* error metrics and run time improvements for MAXDIFF = ∞, 2% and 0.01%. 81

Figure 5.1: The proposed IC-OPC flow. 85

Figure 5.2: The IC-OPC mask synthesizer algorithm. 87

Figure 5.3: Decision matrix for mask correction. 88

Figure 5.4: Effect of mask size changes on the printed *Ion* and gate area of an isolated transistor. 88

Figure 5.5: Mean and standard deviation of gate area deviation and *Ion* deviation for ISCAS85 test circuits. 90

Figure 5.6: Histogram of path delay deviation for post-OPC c1908 circuit. 92

Figure 5.7: Trendline relationship between the mean gate area deviation and the designed transistor width. 92

Figure 5.8: The histogram of path delay deviation for c432, c499, c880 and c6288..93

Figure 6.1: Formulation of PB-OPC mask optimization problem. 100

Chapter 1

Introduction

1.1 Background

Since the early days of microelectronics industry, optical lithography has been the mainstream technology for volume manufacturing in Integrated Circuits (IC) fabrication [1]. The lithography process is the most critical part of IC fabrication, accounting for one-third of the IC manufacturing costs [2, 3] and being the technical limiter for further reduction in transistor size [4].

The steps in the lithography process are shown in Figure 1.1 [5]. First, a small volume of liquid resist is dispensed onto a wafer. This is then followed by spinning the wafer at high speed to fling off the excess resist and allow the solvent to evaporate. The residual solvent inside the resist film is further removed via bake-induced evaporation in a soft bake operation. In the exposure step, the resist-coated wafer is exposed to a pattern of intense light (which is an image formed on the wafer inside the projection exposure system shown in Figure 1.2). After exposure, a post-exposure bake is performed to stimulate chemical reaction to alter the resist solubility characteristic [6]. Subsequently, only the exposed resist areas of positive resist type (or unexposed resist areas of negative resist type) are selectively removed during the chemical development step. Finally, the wafer with developed resist is baked to enhance its etching stability. In a typical IC fabrication process, the aforementioned lithography steps could be repeated up to 30 times [5].

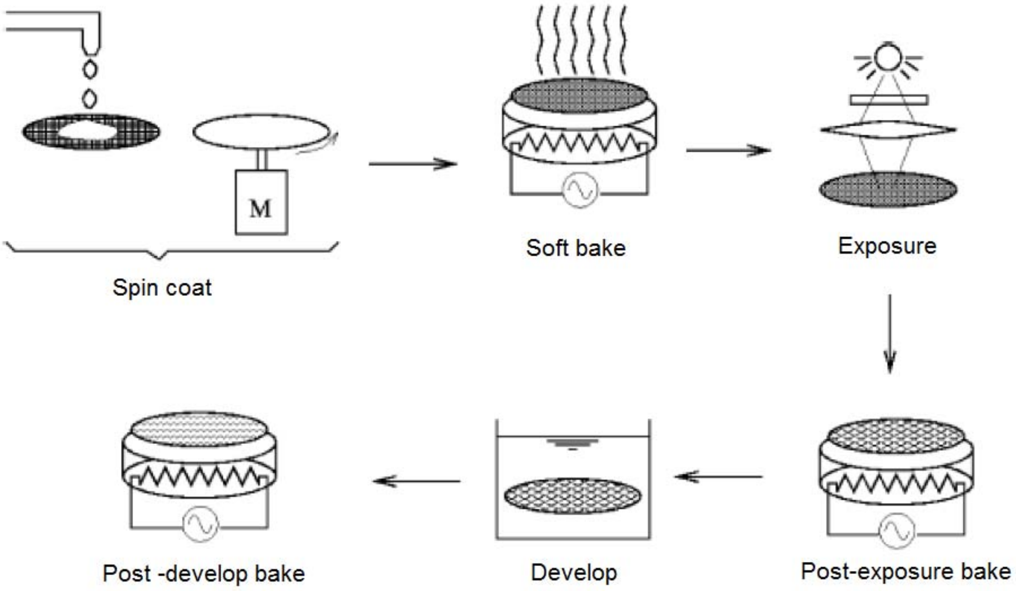


Figure 1.1: Typical steps in the lithography sequence [5].

The commonly used projection exposure system is shown in Figure 1.2 [7]. The operation sequence begins with properly positioning the wafer at the focus level. Then, a shutter in the illumination system is opened to allow light shines through the entire mask in a step-and-repeat wafer stepper. The pattern on the mask is imaged by the lens onto the wafer. This image is reduced laterally by lens reduction factor of 4:1 or 5:1. Large lens reduction factor is desirable because the effects of variations in line widths, misregistration and defect on the mask is reduced by the lens reduction factor [6].

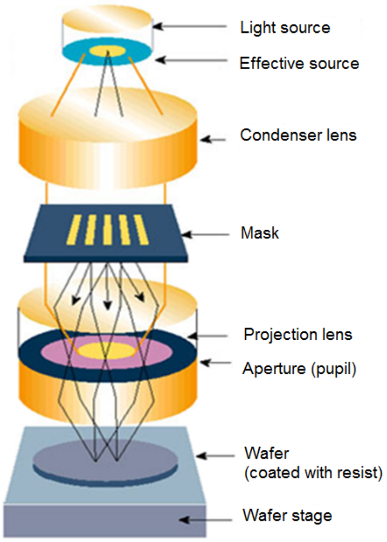


Figure 1.2: Projection exposure system. [7]

The main goal of lithography process is to successfully transfer the patterns from designed IC layout to respective layers on a wafer, within the stringent requirement of critical dimension (*CD*) and overlay control. *CD* is the minimum half pitch resolvable for a diffraction-limited optical projection system. It can be described by the Rayleigh equation as follows:

$$CD = \frac{k_1 \lambda}{NA} \tag{1.1}$$

where k_1 is a process dependent factor determined by resist capability, tool control, mask pattern adjustments and process control [8]. λ is the illumination light wavelength, and *NA* is the numerical aperture of the optics lens. Traditionally, the way of printing smaller *CD* is by using smaller illumination wavelength λ and optics of higher *NA* rating. However, these systems are often developed at a much slower pace than the speed at which *CD* shrinks. Hence, this motivates the development of Resolution Enhancement Techniques (RETs) that can improve the aerial image quality and thereby decrease the k_1 factor to print smaller *CD* [8, 9].

RETs exploit the three variables of electromagnetic wave, namely *amplitude, phase* and *propagation direction* to provide resolution enhancement. The three main approaches in RETs correspond to the control of these three variables: OPC for the wave amplitude, phase-shifting masks for the wavefront phase, and off-axis illumination for the wave direction. Among these approaches, OPC is noted as one of the key technologies enabling 90nm production [10]. It is also a major contributor to the mask costs and mask design turnaround time in lithography [11]. Table 1.1 shows the progression of OPC to extend optical lithography [12]. It becomes much more difficult and expensive to implement OPC at each successive technology generation [8]. Therefore it is of immense interest to research new techniques to reduce the cost of OPC.

Table 1.1: Various techniques for achieving desired *CD* control and overlay with optical projection lithography [12]

MPU M1 contacted ½ pitch	65nm	54nm	32nm	22nm
k_1 range [A]	0.31-0.40	0.28-0.31	0.18-0.28	0.14-0.22
Design rules	Lithography friendly design rules		Double exposure compatible design	
Masks	Model-based OPC with vector simulation, sub-resolution assist feature, polarization corrections	All previous approaches + variation of OPC intensity by location in circuit	All previous approaches + dense OPC. and source mask optimization	All previous approaches + Inverse Lithography

1.1.1 Overview of Optical Proximity Correction (OPC)

OPC is one of the mask engineering techniques used to increase layout-to-wafer pattern fidelity. It is basically a technique of pre-distorting the mask patterns such that the printed patterns closely resemble the desired shapes. This is accomplished by compensating mask geometry for known effects which will occur during imaging or subsequent processing. Figure 1.3 shows an example of qualitative improvement brought about by OPC.

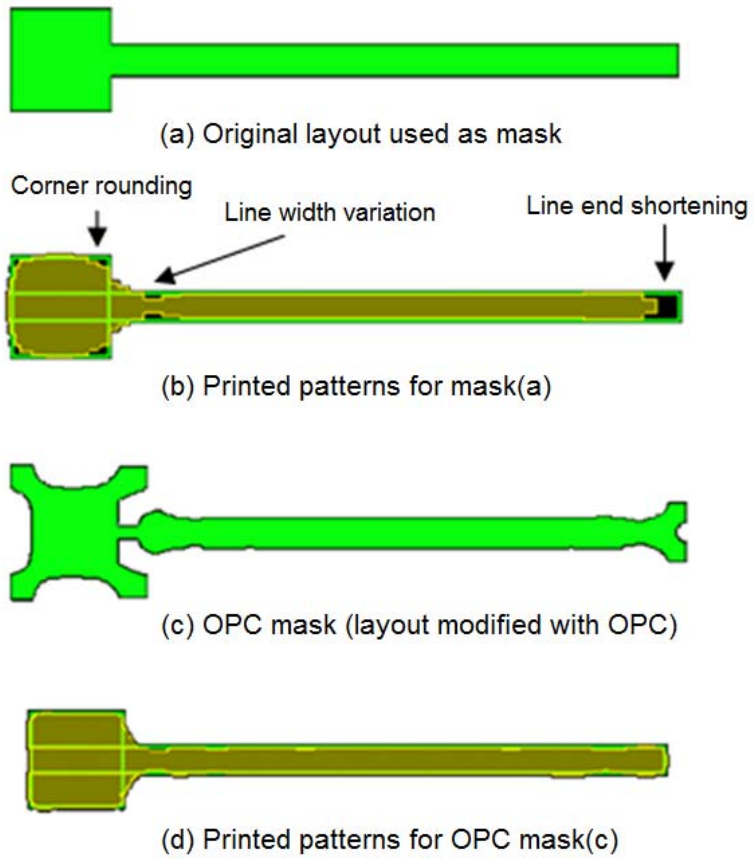
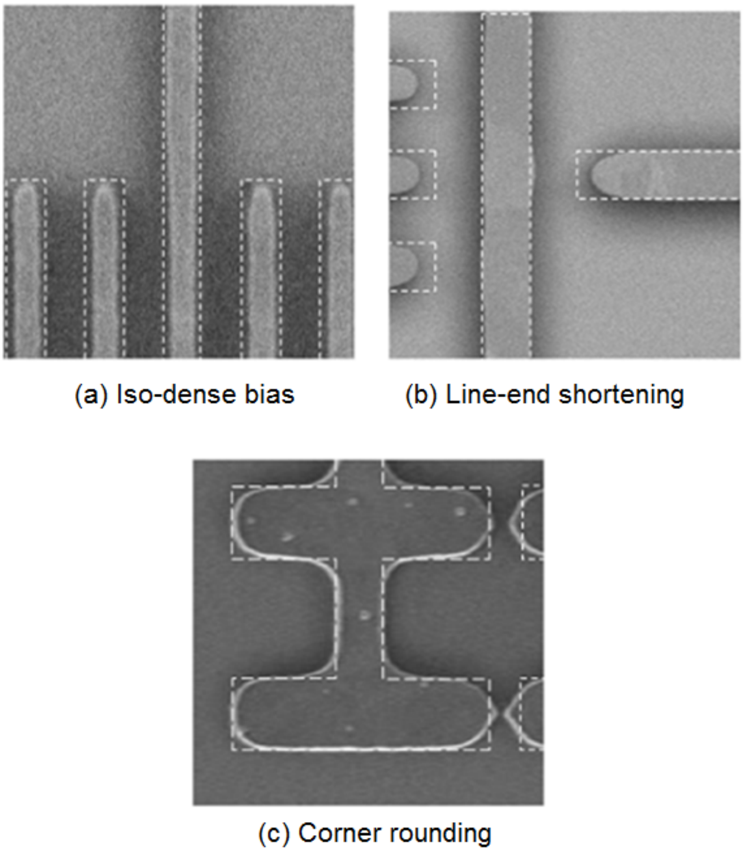

(a) Original layout used as mask
Corner rounding
Line width variation
Line end shortening
(b) Printed patterns for mask(a)
(c) OPC mask (layout modified with OPC)
(d) Printed patterns for OPC mask(c)

Figure 1.3: OPC improves layout-to-wafer pattern fidelity.

Figure 1.4 illustrates the three typical image fidelity problems that can be corrected through OPC – iso-dense bias, line end shortening, and corner rounding

[13]. The iso-dense bias refers to the bias introduced between the isolated and the dense structure as a result of proximity effect. This type of distortion results in across-chip line width variation and can be minimized with selective line biasing method or sub-resolution assist feature insertion during OPC.


(a) Iso-dense bias

(b) Line-end shortening

(c) Corner rounding

Figure 1.4: Typical image fidelity problems in lithography [13].

Another form of image distortion is line end shortening (LES) where the printed length of a rectangle is less than the nominal length. LES results primarily from diffraction, mask pattern rounding, and diffusion of chemical species in resist. As *CD* decreases, LES increases dramatically and negatively impacts both overlay control and circuit density (Figure 1.5 [9]). Figure 1.6 [9] shows the various correction methods used for LES reduction.

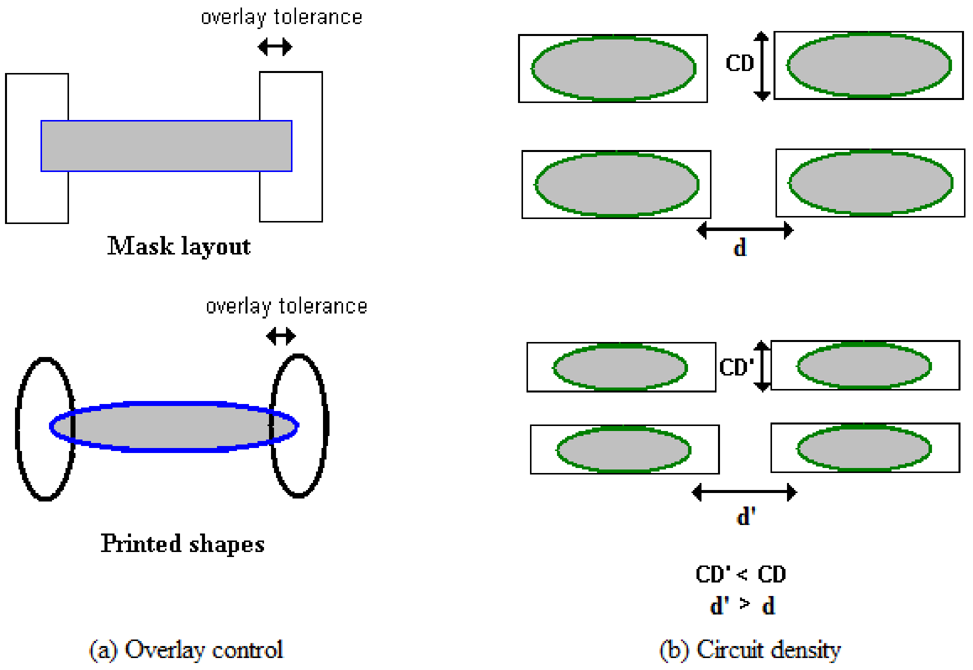


Figure 1.5: Line end shortening impacts overlay control and circuit density [9].

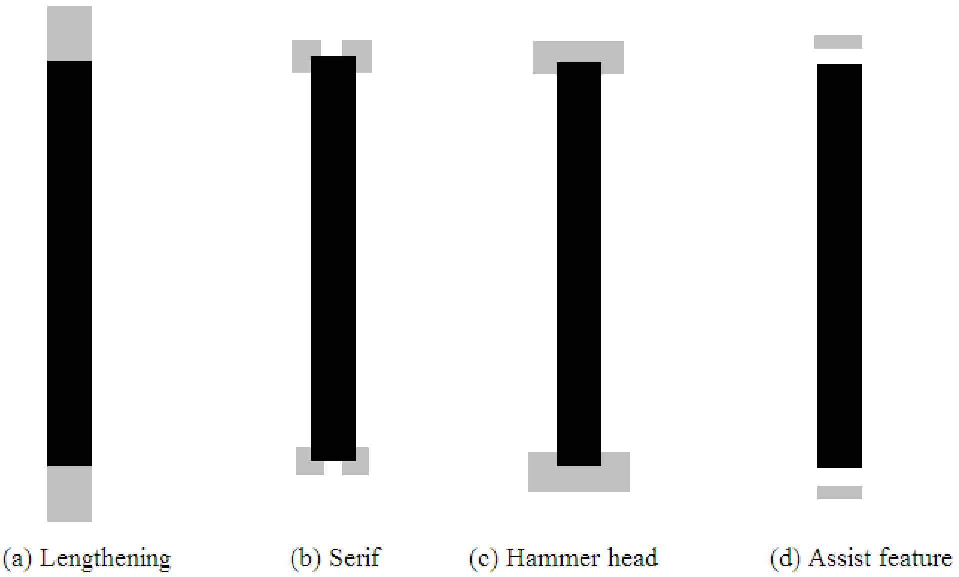
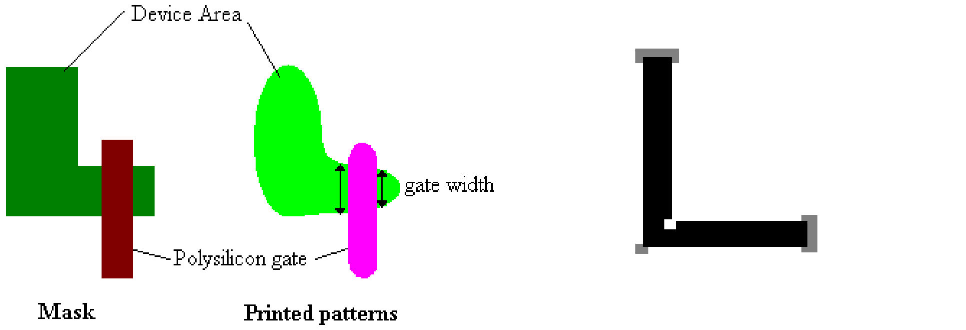


Figure 1.6: Methods for line end shortening reduction [9].

The third form of image distortion is corner rounding, which is inevitable as it is caused by the high frequency components of a sharp corner filtered out by the pupil. Figure 1.7 (a) shows an adverse effect of corner rounding in which the rounding of L-shaped active area elbow results in a device whose effective gate width is dependent on the relative placement of polysilicon gate and active area. As shown in Figure 1.7 (b), corner rounding is generally addressed using serif and antiserif.



(a) Adverse effect of corner rounding (b) Using serifs to reduce corner rounding

Figure 1.7: Corner rounding.

1.1.2 Historical Perspectives of OPC

OPC has been used in IC manufacturing in different forms for many years. Back in the 1970s, circuit designers manually added OPC corrections to the extremely dense circuitry [14]. Serifs were added to the mask, by trial and error, until the desired patterns were successfully printed on the wafer empirically. However, this manual approach is costly, time-consuming and complex. Thus, it is impractical for use in the very large scale IC design. Hence, automated algorithms are needed to improve the efficiency and to enable the fast processing of complex chips.

The various OPC algorithms found from literature could be categorized into two groups: *rule-based OPC* and *model-based OPC*. Rule-based techniques attempt the correction using geometric rules pre-formed by experiment or simulation. A pattern recognition algorithm is used to match a specific geometry to the corresponding prescribed correction. Such an approach is fast, though it is likely to be inaccurate because the correction is not based on real-time lithography simulation. Otto et al. [15] used simulation and supplementary experimental data to generate the geometry correction rules for subsequent rule-based approach. Newmark [16] formed a library of pre-computed corrections to selected patterns using iterative model-based algorithm and the mask corrections are subsequently interpolated from the library.

In contrast, model-based OPC adjusts the corrections based on real-time lithography simulation. The mask edges are moved until the printed patterns are close to the designed layout. Inherently, model-based OPC is generic and more accurate when compared to rule-based OPC. Another advantage of model-based OPC over rule-based OPC is its ability to capture all phenomena (primary and secondary effects) originating from physics incorporated in the models. However, model-based OPC requires much longer computational time than rule-based OPC, mainly due to the time-intensive lithography simulation step.

There are two main approaches to implementing model-based OPC. In the backward approach, the desired printed pattern serves as the starting point and the inverted process model is then used to obtain the optimized layout. Liu and Zakhor [17, 18] proposed pixel-mask model-based optimization method but it was deemed impractical due to the time-consuming and mask designs complication issues. Their subsequent work [19] addressed the mask complexity issue but not the timing.

In the forward model-based OPC approach, the original layout is iteratively modified until the correction is acceptable, both in terms of lithography performance and mask manufacturability. Rieger and Stirniman [20-22] proposed zone sampling and empirically constructed the lumped model of proximity effects for calculating the corrections. Cobb and Zakhor [23-25] formulated a simulation-feedback OPC optimizer as an iterative algorithm involving feedback of correction. Due to its relatively fast simulation time, their proposed work was commercialized for industrial use. Regardless of backward or forward approach, model-based OPC implementation involves correction function derivation and automated mask patterns manipulation by a computer-aided design (CAD) system.

Figure 1.8 shows the forward model-based OPC flow proposed by Cobb and Zakhor [23-25]. Given the input of designed layout, the mask polygon edges are first segmented into independent fragments. The initial mask (which is the replica of designed layout) is subjected to lithography process simulator to generate the wafer print image. The image errors between designed layout and wafer print image are tabulated as edge placement errors (EPEs) data. EPE is defined as the displacement error between the desired layout edge and the printed shape edge at predefined sites [14, 26]. The tool then corrects the EPE by moving individual fragments with calculated resize step and specific direction (inward or outward) based on the local EPE value. The mask fragment correction process is repeated until the EPEs are minimized or reach maximum corrective iterations. The modified mask layout, which is known as EPE-OPC mask, is output to the user. In general, the complexity of the EPE-OPC mask is highly correlated with the fragmentation scheme: larger fragmentation length results in less aggressive

OPC correction and thus less complex mask. Figure 1.9 illustrates the mask complexity for the case of no OPC, medium aggressive OPC and aggressive OPC scheme.

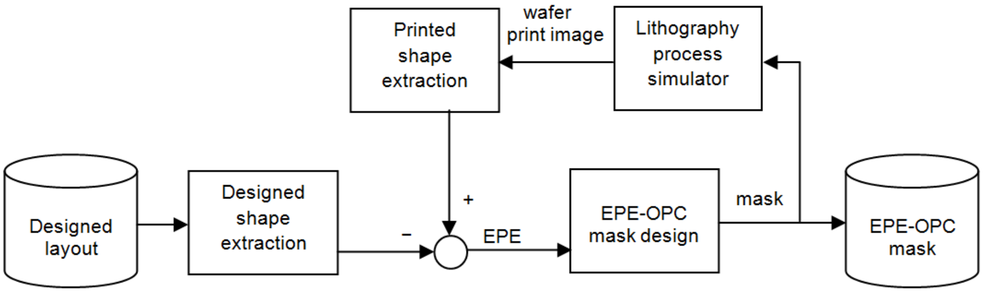
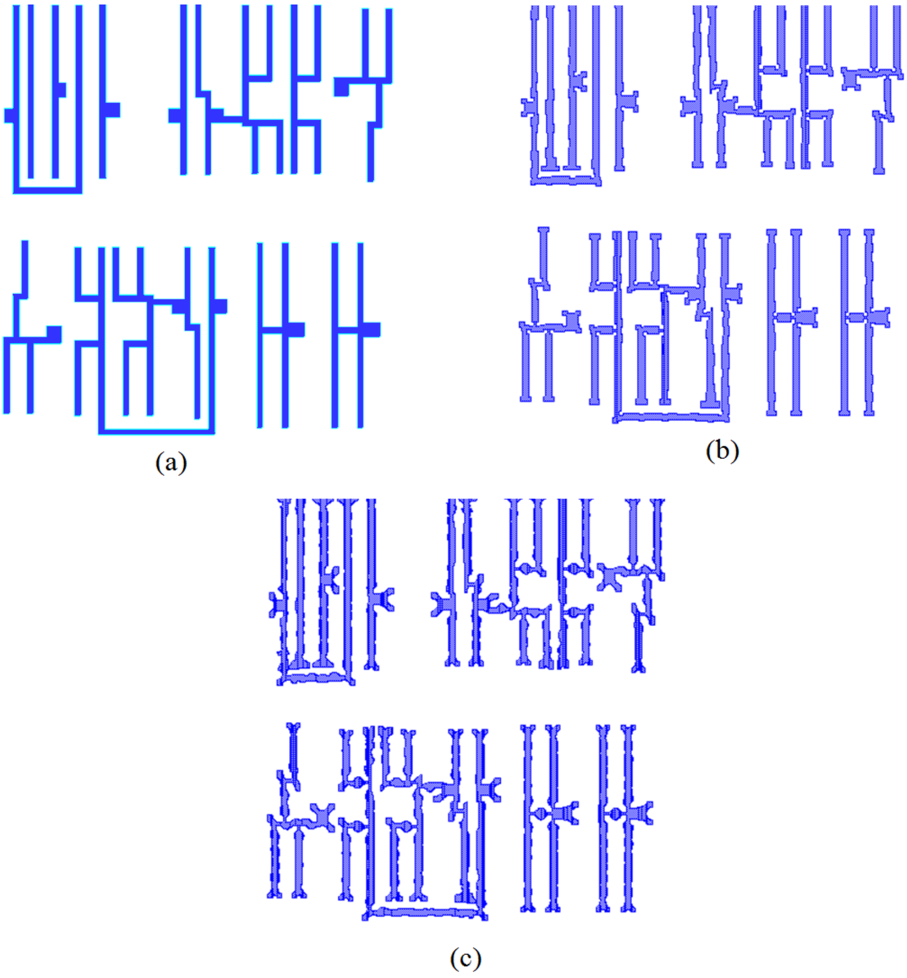


Figure 1.8: Simplified diagram for the forward model-based OPC flow.


(a)

(b)

(c)

Figure 1.9: Mask with (a) no OPC (b) medium aggressive OPC (c) aggressive OPC scheme.

1.1.3 Challenges and Motivation

Overall, OPC is an important step in today's IC manufacturing and has become an integral part of the Design-to-Manufacturing tape-out flow. It is widely used in industry to correct systematic and stable within-field patterning distortions caused by proximity effects so as to minimize the across-chip line width variation [9, 27, 28]. The advent of nano-device makes aggressive OPC correction scheme inevitable in the sub-wavelength printing regime. This directly translates to substantial increase in mask cost as well as the more difficult inspection in the OPC-corrected mask [27-29]. This is because the key cost driver to the mask cost (e.g. mask writing time, defect inspection and repair, and mask data preparation) are proportional to the OPC mask size [27, 30]; and the OPC insertion had caused substantial increase in mask data volume in recent design [31, 32]. The exponential increases in the mask cost with the advanced technology node [27, 33, 34] also includes the higher Non-Recurrent Engineering (NRE) cost, which tends to dominate the total manufacturing cost for low-volume application specific integrated circuit (ASIC) chips production. For 90nm ASIC designs, the mask cost amounts to 60% of the total cost of lithography [28, 35]. This could hinder the ASIC design from leveraging the most advanced CMOS technology to improve their circuit performance. Thus, it is of great interest to develop mask cost-aware OPC solutions.

As mentioned earlier, conventional OPC approach is geometrically EPE driven [11, 14, 36] and tries to match the printed pattern to the designed layout. The impact of the OPC edge insertion on circuit performance is not considered during the OPC correction routine. Therefore, it is possible that an over-corrected

OPC mask would just slightly outperform a moderately-corrected OPC mask but at a much higher cost. Hence, there is a need to incorporate the design intent (circuit performance) into the OPC flow to avoid the above-mentioned scenario. In [11], circuit performance is incorporated into OPC, where the tolerable EPEs were predetermined from the timing analysis and the problem was solved as a constrained OPC insertion with geometry matching. However, the mask cost saving is still limited to these non-critical nodes. The EPE-OPC approach based on objective of minimizing error in the current, rather than the EPE was also proposed in [37, 38]. However, the mask complexity correlates to the fragmentation scheme used and the performance variation minimization are limited as only polysilicon edge fragment movement is permitted in the approach. The impacts of OPC and other lithography-induced imperfectness such as lens aberration and flare on the circuit performance have also been studied empirically and theoretically via various proposed evaluation methodologies [39-43]. Specifically, the circuit performance variability under different OPC settings were analyzed off-line to quantify the different OPC dissection algorithm [43]. A unidirectional link was established to connect the OPC settings to post-OPC circuit performance but not otherwise. This motivates our work to complete the loop by feedback the post-OPC circuit performance and develop a performance-driven OPC algorithm to minimize the performance variation for a given design intent.

Overall, the objective of this research work is to employ design-process integration concepts in the mask design problem so that to provide a cost-effective solution to meet the future device manufacturing requirements. The commonly used benchmark circuits, IEEE International Symposium on Circuits and Systems 1985 (ISCAS' 85) [44-52], are used as the test vehicles to investigate the effectiveness of the proposed solutions.

1.2 Contributions

This thesis presents the development and analysis of circuit performance driven OPC frameworks for mask costs reduction and circuit performance matching improvement. The key contributions of the thesis are listed below.

1.2.1 Design-process Integration for Performance-based OPC (PB-OPC) Framework

A design-process integrated performance-based OPC framework is developed in Chapter 2 to reduce the OPC mask complexity without compromising the overall circuit performance. Involving the integration of commercial lithography simulator and SPICE simulator, the framework is formulated as a negative-feedback system to control the printed transistor performance via iterative knowledge-based mask correction. The proposed framework relies on the estimation of post-lithography transistor performance via the look-up SPICE-based table approach. Then, the mask generation algorithm is designed to alter the mask accordingly to minimize the performance error and mask cost.

The feasibility of the proposed framework is demonstrated via simulation results by comparing its performance against a commercial OPC tool. In the simulation, the post-OPC circuit performances are evaluated based on the equivalent gate lengths of the printed circuits. The simulation results reveal that the proposed framework outperforms the conventional EPE based OPC approach in two aspects: reduction in mask data volume and circuit performance variation. A consistent improvement in the mask complexity and circuit performance has been observed over the various test cases.

1.2.2 Device Performance-based OPC (DPB-OPC) Methodology

Further improvements in the proposed performance-based OPC framework are made.

- For performance extraction, the employed gate-slicing model [53] assumes uniform current density along the device width direction. However, the detailed TCAD simulation [54, 55] revealed that the threshold voltage variation and edge effect could result in non-linear current density along the channel width direction. To account for such effects, a weighting function $\gamma_k(w)$ is augmented to the gate-slicing model used in the framework.
- A rather different mask design algorithm is developed in this improved framework. First, the initial mask adjustment step (*init_adjust*) is pre-characterized using a minimum-sized transistor layout constructed based on the design rule. Then, the characterized *init_adjust* that results in minimum performance deviation error is mapped into a look up table, as a function of transistor channel length and width. Through such systematic approach, the performance deviation error is further reduced by an average of 3.07% reduction in mean *Ion* deviation when compared to the earlier framework.

- A modular block called DRC compliance regulator is implemented in this improved framework to ensure that the post-OPC printed patterns do not exhibit bridging, pinching, open or short issues even in the presence of mask misalignment. As far as the diffusion and polysilicon layers are concerned, the relevant failure mechanisms are bridging between transistors, bridging between polysilicon to neighboring contacts, line-end pull back with overlay errors and insufficient enclosure of contact. The detection and elimination of these failure mechanisms are achieved by monitoring the transistor counts, larger polysilicon to diffusion extension for printed shapes to ensure minimum extension margin (>overlay errors), and larger enclosure of contacts for both polysilicon and diffusion layers if necessary.

.

The improved performance-based framework outperforms the performance-optimized EPE-OPC approach in two aspects: an average of 34% reduction in mask size and up to 13.5% reduction in device performance deviation.

1.2.3 Library-based Device Performance-based OPC for Hierarchical Circuits

The proposed performance-based OPC framework has showed promising results in achieving considerable mask data saving as well as improved circuit performance matching. Despite this, the performance gain is limited by the comparatively longer run time. Due to the iterative performance evaluation of every transistor, the performance-based OPC run time is anticipated to increase exponentially with number of transistors. Therefore, full chip performance-based OPC approach is inefficient for application on the very large scaled integrated (VLSI) circuit comprised of billions of transistor.

To improve the run time efficiency of full chip performance-based OPC, a library-based performance-based OPC methodology for synthesized VLSI circuits is developed. Basically, the synthesized VLSI circuit composed of various standard cell layouts from the provided foundry libraries. By first pre-characterizing the performance-based OPC mask for each standard cell during the library database construction, the entire full chip OPC mask can then be formed by stitching the respective cells' OPC mask per synthesized placement order and thus results in shortened computational time. However, the non-negligible optical proximity effects introduced by boundary cells, especially evident around the cell boundaries region, could contribute to different printing result between the library-based OPC and conventional model-based OPC. This in-turn results in performance disturbance to the transistors at the boundary regions. Such performance disturbance is then rectified by the localized DPB-OPC refinement until the post-placement *Ion* error locally minimized.

Simulation demonstrates that the library-based performance-based OPC approach achieves comparable performance to full chip PB-OPC with significant run time reduction (~44x with ISCAS'85 benchmark design). In addition, better performance matching is achieved in most test cases with library-based performance-based OPC approach. Based on the simulated performance disturbance map, the transistors with degraded *Ion* error can be fine-tuned by the adaptive correction step but at the expense of additional computational effort.

1.2.4 Device Current and Capacitance Oriented OPC (IC-OPC)

As described in Section 1.2.2 [56], a DPB-OPC framework was presented to synthesize simpler masks with printed patterns' *Ion* performance closely matches the designed value. However, DPB-OPC suffers larger gate capacitance deviation than the performance-optimized EPE-OPC, which might result in delay mismatch. In order to achieve better delay matching while reducing the mask complexity, an improved OPC approach namely IC-OPC was proposed in Chapter 5 to consider both post-lithography device drive current and gate capacitance during the correction phase. By simulation, the proposed IC-OPC outperforms the performance-optimized EPE-OPC approach in three aspects: an average of 32% reduction in mean path delay deviation, an average of 34% reduction in mask size and at least of 84% run time saving.

1.3 Organization

This thesis is organized as follows. Chapter 2 describes the design-process integrated PB-OPC in detail. The simulation results are presented to verify the feasibility and effectiveness of the proposed framework. Chapter 3 presents the generalized DPB-OPC framework with three new features. For synthesized VLSI digital circuit, library-based performance-based OPC approach is proposed in Chapter 4 to improve the run time efficiency of the previously developed framework. Chapter 5 introduces IC-OPC framework for co-optimizing post-OPC circuit performance (*Ion*, gate capacitance and delay) and mask complexity. Finally, conclusion and future work are provided in Chapter 6.

Chapter 2

Design-process Integration for Performance-based OPC (PB-OPC) Framework

2.1 Introduction

OPC is an integral part of the Design-to-Manufacturing tape-out flow. It is widely used in the industry to correct systematic and stable within-field patterning distortions caused by proximity effects to minimize the across-chip line width variation [9, 27, 28]. The advent of nano-device makes aggressive OPC correction scheme inevitable in the sub-wavelength printing regime. This directly translates to substantial increase in mask cost as well as the more difficult inspection in the OPC-corrected mask [27-29]. This is because the key cost driver to the mask cost (e.g. mask writing time, defect inspection and repair, and mask data preparation) are proportionate to the OPC mask size [27, 30]; and the OPC insertion has caused substantial increase in mask data volume in recent design [31, 32]. The exponential increases in the mask cost along with the advanced technology node [27, 33, 34] also includes the higher NRE cost, which tends to dominate the total manufacturing cost for low-volume ASIC chips production. For 90nm ASIC designs, the mask cost amount to 60% of the total cost of lithography [28, 35]. This could hinder the ASIC design from leveraging the most advanced CMOS technology to improve their circuit performance.

Investigation of current design-manufacturing interface revealed that the conventional OPC methodology is geometrical based [11, 14, 36], which tries to minimize the edge placement errors (EPEs) over correction iterations. During the correction, there is no linkage established between the OPC mask changes and the resulted circuit performance shift. Hence, it is possible that an over-corrected OPC mask would just slightly outperform a moderately-corrected OPC mask but at a much higher cost. In order to avoid such unfavorable correction, the design intent (circuit performance) needs to be leveraged into the OPC flow.

Among the related works in the literature that make sure of circuit performance in OPC, Gupta et al. [11] first formulated the problem as a constrained OPC insertion with relaxed EPE obtained from timing analysis. Banerjee et al. [37, 38] then refined the EPE-OPC approach for the objective of minimizing the error in the electrical current. However, the mask cost saving are limited to the non-critical nodes in [11] and mask complexity correlates to the fragmentation scheme used in [37, 38]. On the other hand, the circuit performance variability under different OPC settings [43] or other lithography imperfectness such as lens aberration and flare [39-43] have also been studied. OPC settings were linked to post-OPC circuit performance but not otherwise. This motivates our work to complete the loop by feedback the post-OPC circuit performance to the OPC algorithm, for minimizing the performance variation for a given design intent. The *in-situ* performance extraction is made feasible via the availability of models to access the device characteristic with distorted device shape [53-55, 57-59].

To demonstrate the concept of performance-based mask generation, a PB-OPC algorithm that will be driven by the transistor performance rather than the desired mask pattern is proposed; with the goal to produce a simpler mask with printed patterns' performance that closely match with the designed value. The two key performance indexes of transistors, *Ion* and leakage current (*Ioff*) can be chosen to monitor this PB-OPC flow. The derived PB-OPC model is targeted for adjusting the polysilicon and diffusion masks which directly define the gate region and affect the device characteristic. Besides, the modeling approach in [53] is employed to estimate the printed transistors' performance such that the Transistor Performance Error (TPE) ~ analogous to EPE in the conventional OPC ~ can be feedback as the mask quality metrics. The implementation of the automated PB-OPC flow is achieved by integrating the lithography process simulator, circuit design tool and algorithm within a Perl script.

By simulation, the proposed PB-OPC outperforms the conventional EPE-OPC in two aspects: at least of 33% reduction in mask MEBES size and 11 to 97% reduction in circuit performance variations. In addition, the PB-OPC framework is applicable to any generic CMOS circuit because of its underlying principle - optimize the mask by matching individual transistor's current characteristic and fairly assumed that the overall circuit performance would also collectively match with the designed value.

This chapter is organized as follows. Section 2.2 describes how the proposed PB-OPC flow works. Section 2.3 discusses the simulation results in which the performance of post-PB-OPC and post-EPE-OPC circuit are compared. Finally, chapter summary is presented in Section 2.4.

2.2 The Proposed PB-OPC Framework

The proposed PB-OPC framework is shown in Figure 2.1. Inputs are the designed layout and the device characteristic library (DCLib). From the given design layout, the desired performance of individual transistor is first extracted according to its gate region dimension. The mask is then initialized to an exact replica of the given design layout and subjected to the lithography process simulator (Mentor Graphics Calibre Workbench [60]) to generate the resulting printed patterns on the wafer. Based on these printed patterns, the performance of individual transistor is extracted and feedback to the mask generation controller by form of *TPE.* The mask is then iteratively modified until the *TPE* becomes locally-minimized. It should be pointed out that only simple geometry alteration, such as stretching or compressing the masks of the original design that change the related transistor gate region, is adopted to minimize the mask production cost. Final output is a simpler PB-OPC mask which will result in printed patterns that match the designed circuit performance. The detailed explanations of the building blocks are presented as follows.

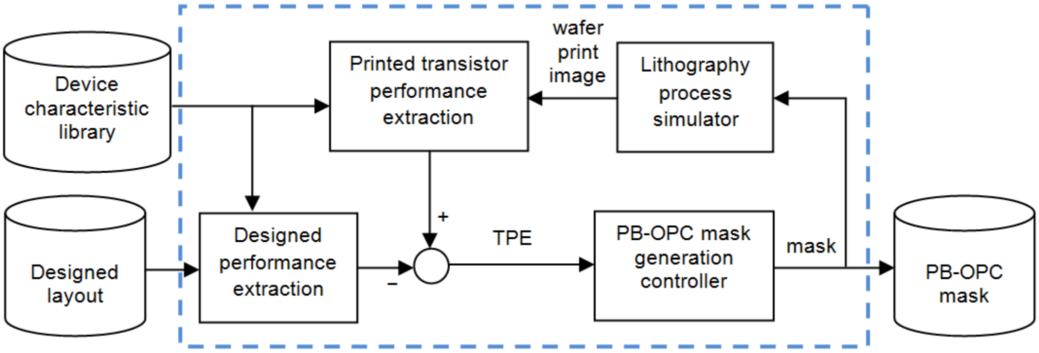

Device characteristic library
Designed layout
Printed transistor performance extraction
wafer print image
Lithography process simulator
Designed performance extraction
+
-
TPE
PB-OPC mask generation controller
mask
PB-OPC mask

Figure 2.1: Flowchart of the proposed PB-OPC framework.

2.2.1 Device Characteristic Library

The DCLib consists of four look-up tables – *Ion* and *Ioff* for wide transistors NMOS and PMOS with different channel length (*L*). One key assumption that validates the modeling approach in [53] is that the characterized transistor width has to be wide enough such that the obtained current characteristic has negligible width-dependency. The 65nm DCLib was created based on the SPICE simulation result [61] using channel width of 10µm and 65nm BSIM4 model card [62].

2.2.2 Designed Performance Extraction

The gate region dimensions for each transistor, i.e. the channel length *L* and channel width *W* of NMOS or PMOS, are extracted from the designed layout. For each transistor, the designed *Ion* and *Ioff* are approximated as:

$$I_{j,design} = I_{wide}(L_{j,design}) \times \frac{W_{j,design}}{W_{wide}} \tag{2.1}$$

where j = index of the transistor

2.2.3 Lithography Process Simulation

To capture the optical proximity effect on the on-wafer printed patterns within the proposed framework, the Mentor Graphics Calibre Workbench [60] is used as the lithography process simulator. Based on the user specified optical and process model, Calibre Workbench simulates the on-wafer printed patterns and outputs it as rectilinear polygons. The optical model includes parameters such as exposure wavelength, partial coherence factor, numerical aperture, illumination scheme and film stack. The process model consists of aerial image parameters such as intensity, image slope and maximum intensity. Optimized and well-calibrated optical and process models are capable of characterizing the lithographic systematic distortions. Hence, the on-wafer printed patterns variation due to these systematic distortions can be corrected. In this simulation work, optical model (model parameters: wavelength (λ) = 193nm, partial coherence (σ) = 0.75, numerical aperture (NA) = 0.75, and standard illumination) and the default Variable Threshold Resist (VTR) process model are used.

2.2.4 Printed Transistor Performance Extraction

The proposed PB-OPC flow relies on the “in-situ” estimation of the post-lithography transistor performance at each iteration. Figure 2.2 shows an example of on-wafer printed patterns for both polysilicon and diffusion layers. The overlapping area of both layers’ printed patterns defines the printed gate region.

As non-rectangular gates (NRG) are inevitable in today state-of-art nanometer lithography process, several modeling approaches have been proposed to predict the properties of NRG's transistor [53-55, 57-59]. Among these methods, the gate-slicing model proposed in [53] is employed to extract the device electrical performance. The model approximates NRG as a set of independent slice transistors connected in parallel (Figure 2.2). To capture its edge contour, the printed gate region is decomposed into *m* rectangular slices using variable width sampling scheme, each with width W_k and length L_k. Thus the total NRG current I_{total} is the sum of all slice currents.

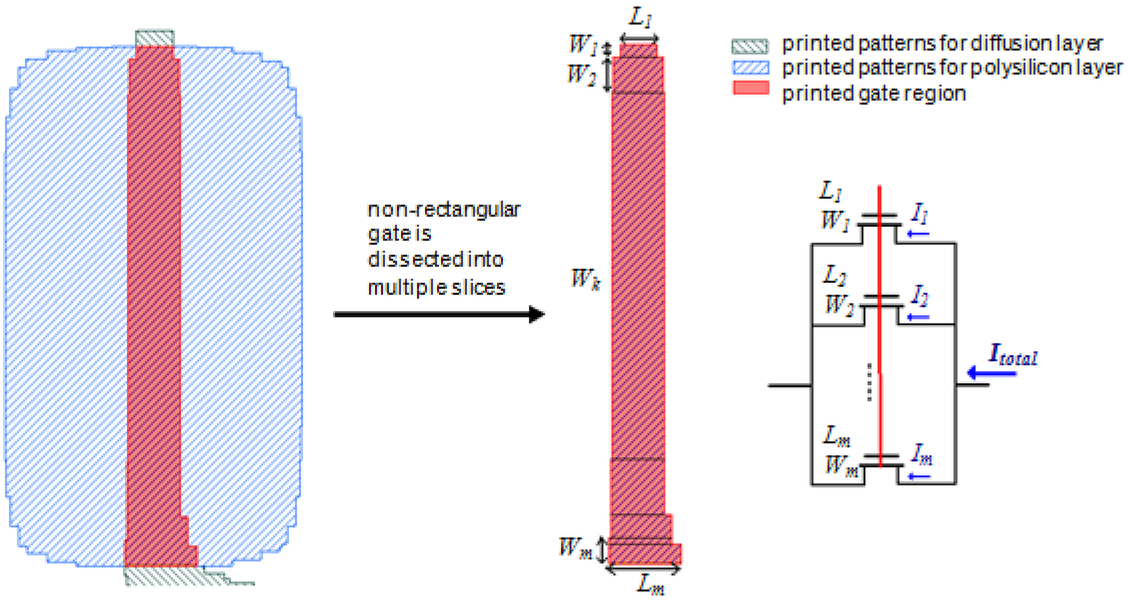
$$I_j = \sum_{k=1}^{m} I_k = \sum_{k=1}^{m} I_{wide}(L_k) \times \frac{W_k}{W_{wide}} \qquad (2.2)$$

where m = total number of slices

k = index of the slice

j = index of the transistor

The total current of non-rectangular transistor can then be used to extract an equivalent channel length based on *Ion* (i.e. Leq_{Ion}) or *Ioff* (i.e. Leq_{Ioff}). The Leq_{Ion} value will be updated into the circuit netlist for post-OPC's circuit performances simulation. Figure 2.3 reveals the negligible estimation error when estimating the current characteristic of W_{slice} = 100nm and W_{slice} =1000nm using the 10µm-wide-transistor library.



printed patterns for diffusion layer
printed patterns for polysilicon layer
printed gate region
non-rectangular gate is dissected into multiple slices
L_1
W_1
W_2
W_k
W_m
L_m
L_1
W_1
I_1
L_2
W_2
I_2
I_{total}
L_m
W_m
I_m

Figure 2.2: Performance extraction for nonrectangular gate.

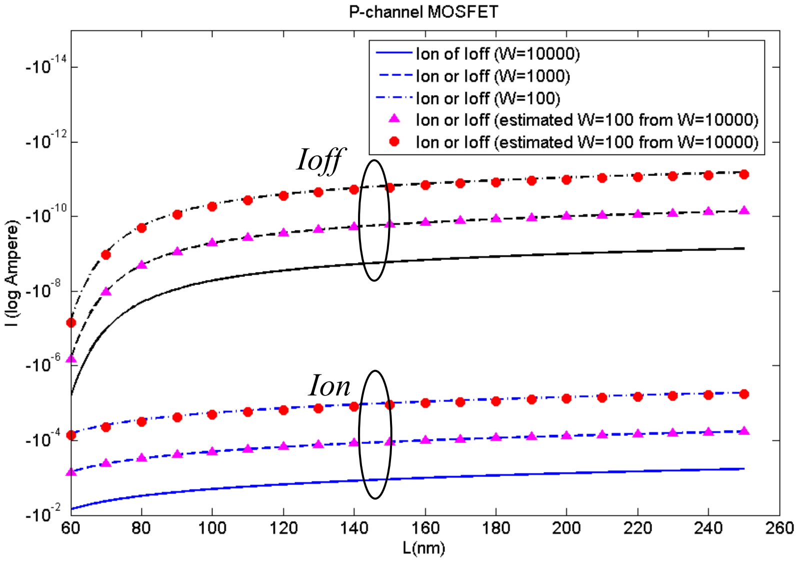


Figure 2.3: Estimation of the transistor’s current characteristic.

2.2.5 Mask Generation Algorithm

As a result of nonlinear pattern transfer during the lithography process, the printed wafer pattern differs from the original layout, causing the printed transistor performance to deviate from the designed value. The performance differences, *TPE* is feedback to the mask generation controller and the control algorithm is designed such that successive mask modification leads to the local minima. *TPE* is defined as following:

$$TPE_{Ion}(j) = \frac{I_{Ion}(j)_{printed} - I_{Ion}(j)_{design}}{I_{Ion}(j)_{design}} \times 100\% \qquad (2.3)$$

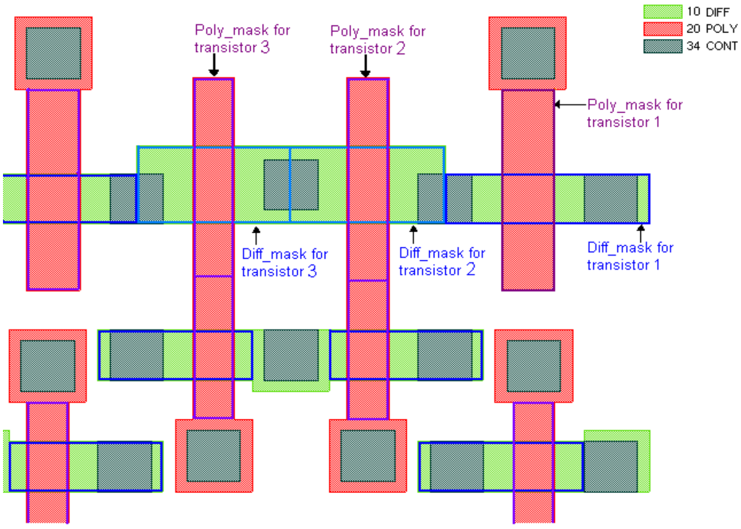
$$TPE_{Ioff}(j) = \frac{I_{Ioff}(j)_{printed} - I_{Ioff}(j)_{design}}{I_{Ioff}(j)_{design}} \times 100\% \qquad (2.4)$$

where j = index of the transistor

In order to produce simpler mask with matched circuit performance, the mask changing algorithm is limited to merely resizing the polysilicon and diffusion mask polygon which define the gate region. For the case where multiple gates are sharing the same diffusion polygon, the segmentation algorithm would divide this polygon into two parts to allow freedom of mask adjustment in *W* direction. Figure 2.4 illustrates the segmentation process and Figure 2.5 displays the flowchart of the mask generation algorithm. The mask correction in *L* or *W* directions is achieved by simply stretching or compressing the segmented polygons. It is performed iteratively until the *TPE* fall within the user-assigned tolerance, or reaches the local minima solution. The local minima solution is guaranteed by the algorithm due to the two conditions:

- The starting point of search is initialized to the vicinity of minima point.
- Local iterative descent method is used to minimize the *TPE* until reaches the local minima point.

In the implementation of the overall PB-OPC process, users are given flexibility to prioritize certain performance goal to suit their design need. The three supported operation modes are explained in Table 2.1. Since the goal is to compare the printed circuit performance between EPE-OPC and the PB-OPC, only Mode_*Ion* will be used in the simulation work to minimize TPE_{Ion}..



🟩 10 DIFF
🟥 20 POLY
▨ 34 CONT
Poly_mask for transistor 3
Poly_mask for transistor 2
Poly_mask for transistor 1
Diff_mask for transistor 3
Diff_mask for transistor 2
Diff_mask for transistor 1

Figure 2.4: Segmentation process.

Table 2.1: The three supported operation modes in PB-OPC.

Mode	PB-OPC goal
Mode_*Ion*	Mask correction algorithm driven by TPE_{Ion} only
Mode_*Ioff*	Mask correction algorithm driven by TPE_{Ioff} only
Mode_*IoffIon*	Mask correction algorithm driven by TPE_{Ioff} and afterwards TPE_{Ion}

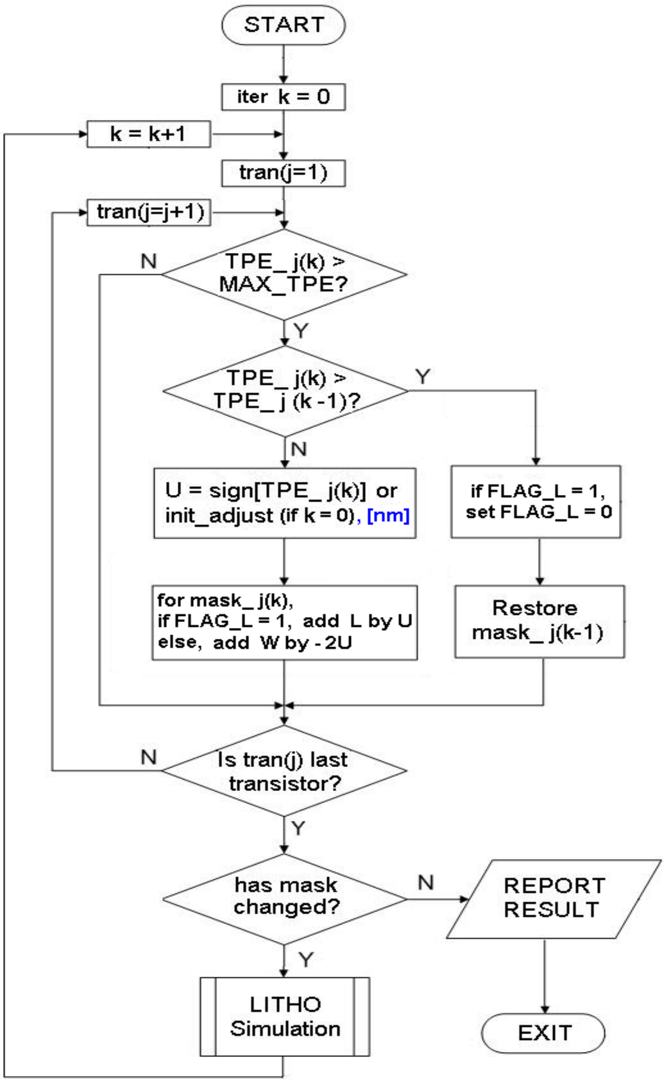

START
iter k = 0
k = k+1
tran(j=1)
tran(j=j+1)
N
TPE_ j(k) > MAX_TPE?
Y
TPE_ j(k) > TPE_ j (k -1)?
Y
N
U = sign[TPE_ j(k)] or init_adjust (if k = 0), [nm]
if FLAG_L = 1, set FLAG_L = 0
for mask_ j(k), if FLAG_L = 1, add L by U else, add W by - 2U
Restore mask_ j(k-1)
N
Is tran(j) last transistor?
Y
has mask changed?
N
REPORT RESULT
Y
LITHO Simulation
EXIT

Figure 2.5: Flowchart of the mask generation algorithm.

2.3 Results and Discussions

The PB-OPC framework is implemented using Perl script for both Linux and UNIX machine. The framework is tested with transistor NMOS and PMOS, some standard digital cells, a six-stage inverter chain and a 4-Bytes 6T-SRAM circuit (32 SRAM cells). Table 2.2 to 2.5 compare the circuit performance, mask cost and simulation time on each of the test circuits between the conventional EPE-based OPC and the proposed PB-OPC approach. Section 2.3.1 to 2.3.4 discuss and analyze the comparison results.

The Mentor Graphic Calibre OPCpro is used to generate the conventional EPE-based OPC masks. However, there are many adjustable parameters which will affect the final shape of the OPC mask, and thus the circuit performance of the printed pattern indirectly. Some important OPC settings are fragmentation length (minedgelength, maxedgelength, concavecorn, conedge), step change and number of iterations. For a fair comparison, an optimal Calibre OPCpro setting based on the closely matched designed performance was first determined. A generic cost function is defined as

$$J = \frac{q_1}{N}\sum_{j=1}^{N} e_{ion}{}^2(j) + \frac{q_2}{N}\sum_{j=1}^{N} e_{ioff}{}^2(j) \qquad (2.5)$$

where q_1 = weighting factor for mean square *Ion* error

q_2 = weighting factor for mean square *Ioff* error

N = number of transistors in the circuit of interest

$$e_s(j) = \frac{I_{s\,printed} - I_{s\,design}}{I_{s\,design}} \times 100\%$$

To solve for the optimal OPCpro setting, the search space is first defined to be bounded by fragmentation length (frag) between 10nm-70nm in step of 5nm, step change between 1nm-2nm in step of 1nm, and number of iteration (iter) between 1-20 in step of 2. The weighting factors $q_1$and q_2 should be selected based on the importance of *Ion* and *Ioff* matching in determining the circuit performance of interest. Higher the weighting factor, higher the error contribution into the cost function. For a high speed design, *Ion* matching is more important; but for a low power design, *Ioff* matching should weight more by choosing higher q2. In this work, the weighting factors, q_1 = 1 and q_2 =0, are selected to reflect the importance of *Ion* matching in minimizing the performance variation. As a good *Ion* matching translates to a closer Leq_{Ion} value to the designed *L,* therefore closer match with the designed performance. The point with minima cost value within the search space will be used as the optimal OPCpro setting to generate the EPE-OPC mask. Figure 2.6 shows an example of the cost function plot in the search space for the 4-Bytes SRAM circuit.

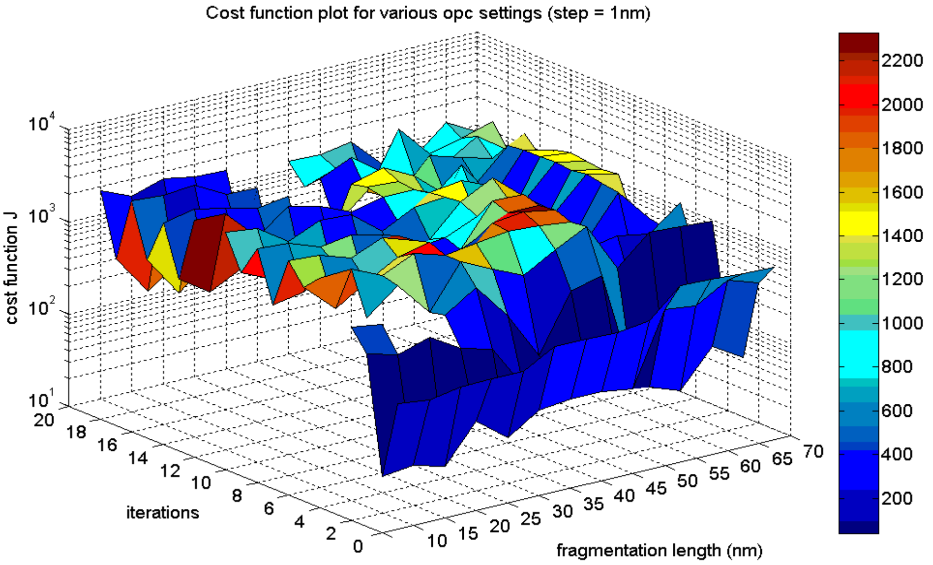
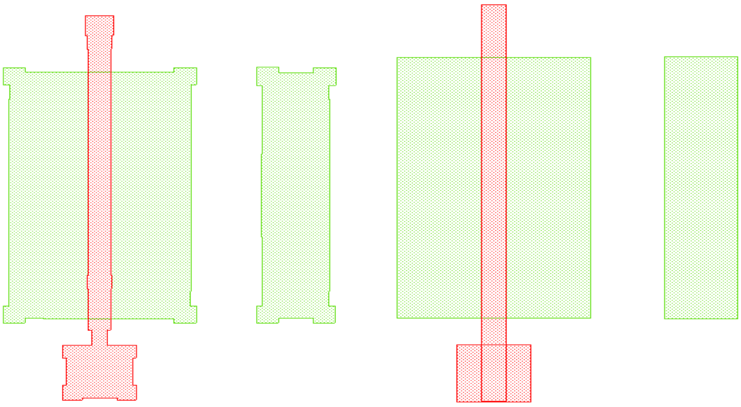


Figure 2.6: Cost function plot within the search space (step = 1nm). Locally optimal OPC setting is frag 20nm, step 1nm, iter 2.

2.3.1 Transistor NMOS and PMOS

Table 2.2 and 2.3 clearly show that PB-OPC achieved at least 59% reduction of current deviation. As illustrated in Figure 2.7, PB-OPC approach also results in a much simpler mask than the conventional OPC mask. The PB-OPC mask is simple because there is no extra edge insertion being introduced by the correction and the segmented mask polygon are just resized accordingly to the *TPE.* A 2000 by 2000 array of single NMOS or PMOS transistor is created and fractured for both approaches. Calibre workbench is employed to fracture the OPC mask from GDSII format to MEBES format (a standard format for e-beam mask writing tool). 55-93% mask volume reduction is achieved in the proposed scheme. This results in shorter mask writing time and less complicated inspection process, and hence is favored if the saved time exceeds the increased OPC run time. The PB-OPC's run time is about 3-4x slower than the commercial EPE based OPC tool due to the integration of different commercial design and process software tool in the proposed OPC flow. Therefore the OPC run time efficiency can be further improved if the overall OPC flow is integrated into single software platform.



(a) EPE-OPC mask (b) PB-OPC mask

Figure 2.7: Comparison between EPE-OPC mask (frag 65nm, step 2nm, iter 3) and PB-OPC mask.

Table 2.2: 65nm NMOS transistor.

OPC approach		EPE	PB	|PB|/|EPE| -1 (%)
Printed circuit performance error (%)	*Ion*	-0.054	0.022	-59
	Ioff	29.24	-1.04	-96
MEBES size for 2000 by 2000 array of NMOS (Bytes)	Polysilicon mask	1865728	841728	-55
	Diffusion mask	3485696	1130496	-68
OPC run time (s)		19.0	86.7	+357

Table 2.3: 65nm PMOS transistor.

OPC approach		EPE	PB	|PB|/|EPE| -1 (%)
Printed circuit performance error (%)	*Ion*	-0.062	-0.018	-71
	Ioff	12.71	-1.12	-91
MEBES size for 2000 by 2000 array of PMOS (Bytes)	Polysilicon mask	4710400	841728	-82
	Diffusion mask	15630336	1130496	-93
OPC run time (s)		19.4	80.1	+412

2.3.2 Standard Digital Cells

The proposed PB-OPC is also tested on some commonly used standard digital cells. Although *Ion* is still used as the controlled output in adjusting the mask, it is more meaningful to look at other performance metrics related to the digital cells. In Table 2.4, the rise time (t_r), fall time (t_f), propagation delay (t_{pLH}, t_{pHL}) of the digital cells are listed for comparison. It can be seen that at least 50% reduction of mask data volume and 80% reduction of performance variation are achieved in the proposed PB-OPC flow.

Table 2.4: Standard digital cells.

OPC approach			EPE	PB	|PB|/|EPE| -1 (%)
Inverter	Printed circuit performance error (%)	t_r	0.18	0	-100
		t_f	0.35	0	-100
		tp_{LH}	-2.86	0	-100
		tp_{HL}	4.20	0	-100
	MEBES size for 1000 by 1000 array of single cell (Bytes)	Polysilicon mask	21035008	6959104	-67
		Diffusion mask	11034624	4767744	-57
	OPC run time (s)		2.60	81.06	+3018
NAND	Printed circuit performance error (%)	t_r	-0.99	-0.04	-96
		t_f	1.12	-0.06	-95
		tp_{LH}	-6.27	0.82	-87
		tp_{HL}	10.44	0.69	-93
	MEBES size for 1000 by 1000 array of single cell (Bytes)	Polysilicon mask	35364864	17504256	-51
		Diffusion mask	19509248	7469056	-62
	OPC run time (s)		2.73	70.80	+2493
NOR	Printed circuit performance error (%)	t_r	4.37	0.09	-98
		t_f	-2.54	0.31	-88
		tp_{LH}	3.23	0.20	-94
		tp_{HL}	22.28	-3.98	-82
	MEBES size for 1000 by 1000 array of single cell (Bytes)	Polysilicon mask	35203072	17369088	-51
		Diffusion mask	26519552	10768384	-59
	OPC run time (s)		2.75	108.61	+3849

2.3.3 Six-stage Inverter Chain

The six-stage inverter chain is optimally designed for minimal propagation delay when driving an external load of 4pF using 65nm technology. Each inverter stage is progressively sized with an effective fan out of 3.5 and the final layout is comprised of 120 transistors using the 65nm technology (Figure 2.8). After subjecting the layout to both EPE-OPC and PB-OPC, Table 2.5 revealed that PB-OPC approach results in less complicated and cheaper mask (about 33-78% reduction in mask MEBES size) with closely matched circuit performance. The overall circuit performance is evaluated by replacing the individual transistor's L in the netlist with Leq_{Ion}. The transient performance (t_r, t_f) and the propagation delay (tp_{LH}, tp_{HL}) are extracted from the SPICE simulation and the PB-OPC mask matches closer to the designed performance.

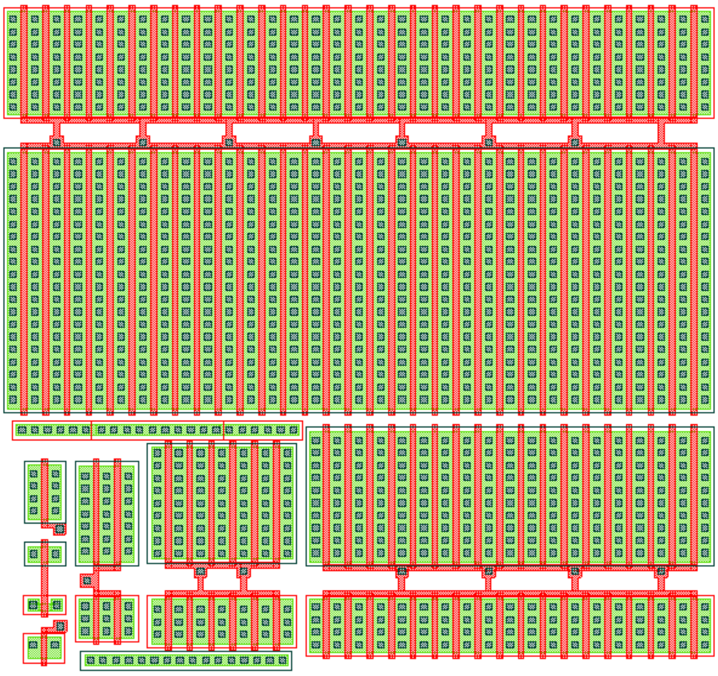
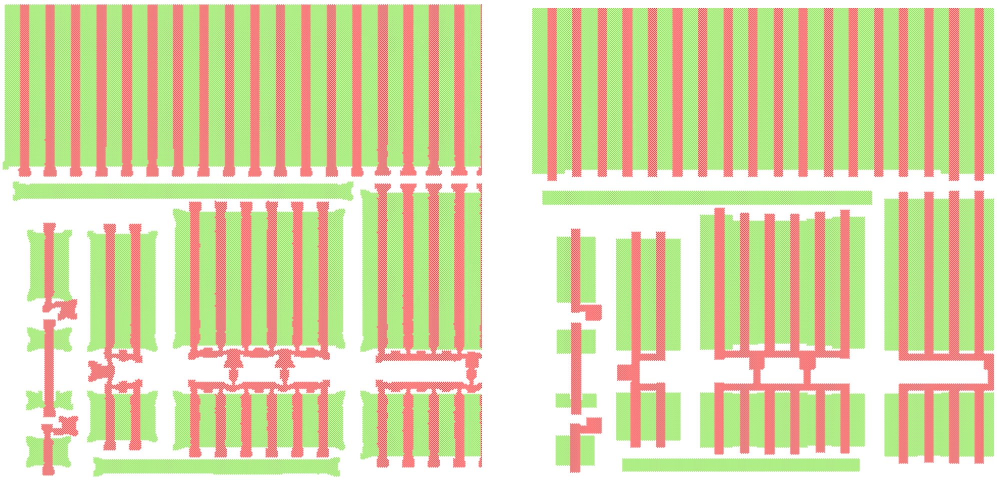


Figure 2.8: Layout of the six-stage inverter chain.

Table 2.5: Six-stage inverter chain.

OPC approach		EPE	PB	|PB|/|EPE| -1 (%)
Printed circuit performance error (%)	t_r	-4.37	-3.88	-11
	t_f	4.23	0.11	-97
	tp_{LH}	-3.70	-1.39	-62
	tp_{HL}	4.54	4.01	-12
MEBES size (Bytes)	Polysilicon mask	18432	4096	-78
	Diffusion mask	6144	4096	-33
OPC run time (s)		66.0	596.6	+804



(a) EPE-OPC mask

(b) PB-OPC mask

Figure 2.9: Comparison between EPE-OPC mask (frag 30nm, step 1nm, iter 4) and PB- OPC mask.

2.3.4 4-Bytes 6T-SRAM Cell

The 6T-SRAM memory cell layout used is based on [63] (Figure 2.10). A 6T-SRAM bit cell is consists of cross-coupled inverter pair and two access transistors. One of the key electrical parameter is Static Noise Margin (SNM), which defined as the minimum dc noise voltage necessary to flip the state. SNM is extracted by a script written based on approach in [64]. Table 2.6 shows over 60% mask size reduction and 90% improvements in SNM performance in the PB-OPC approach. Figure 2.12 revealed that the SRAM butterfly plot of the printed pattern from PB-OPC behaves almost the same as the designed butterfly plot.

Table 2.6: 4-Bytes 6T-SRAM cell.

OPC approach		EPE	PB	|PB|/|EPE| -1 (%)
Printed Circuit Performance Error (%)	SNM (left)	4.24	-0.42	-90
	SNM (right)	3.91	0.18	-95
MEBES size (Bytes)	Polysilicon mask	18432	6144	-67
	Diffusion mask	18432	6144	-67
OPC run time (s)		63.3	235.5	+272

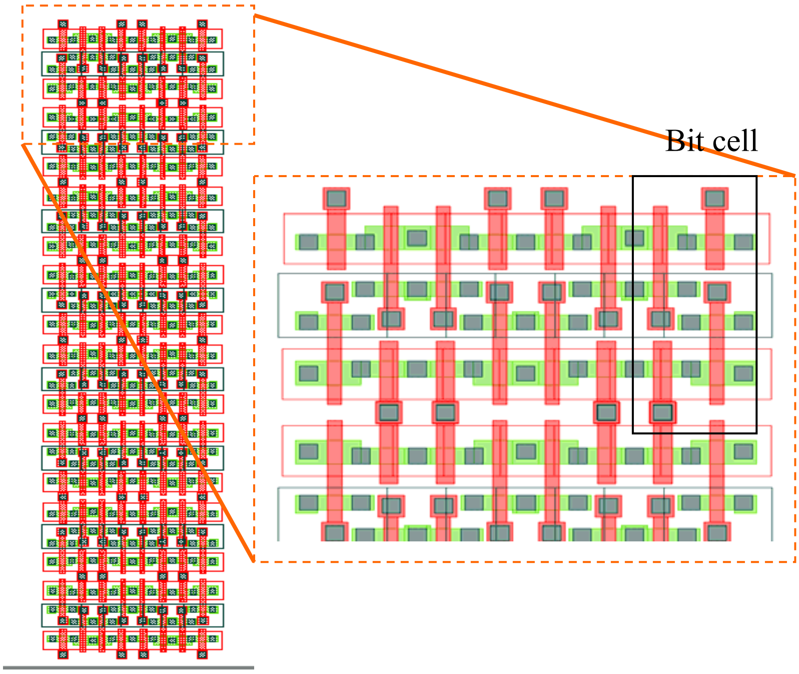
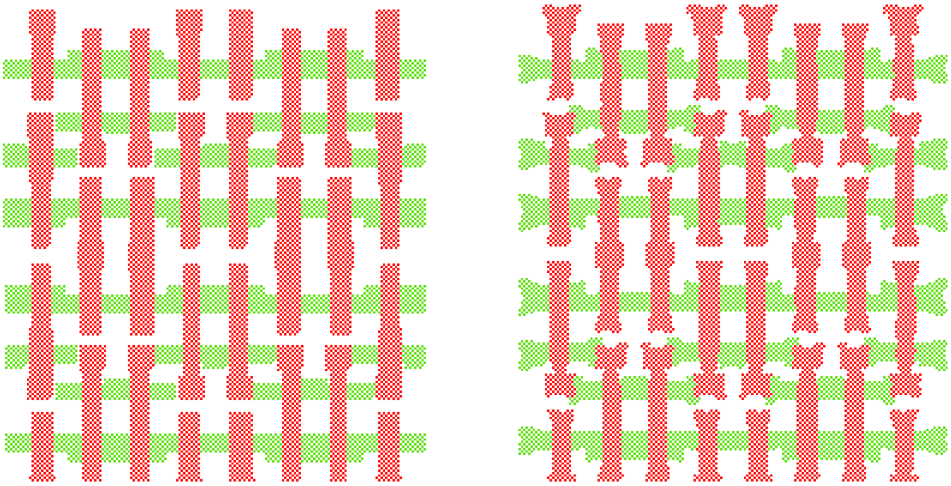


Figure 2.10: Layout of the 4-Bytes 6T-SRAM cell.



(a) EPE-OPC mask (b) PB-OPC mask

Figure 2.11: Comparison between EPE-OPC mask (frag 20nm, step 1nm, iter 2) and PB-OPC mask.

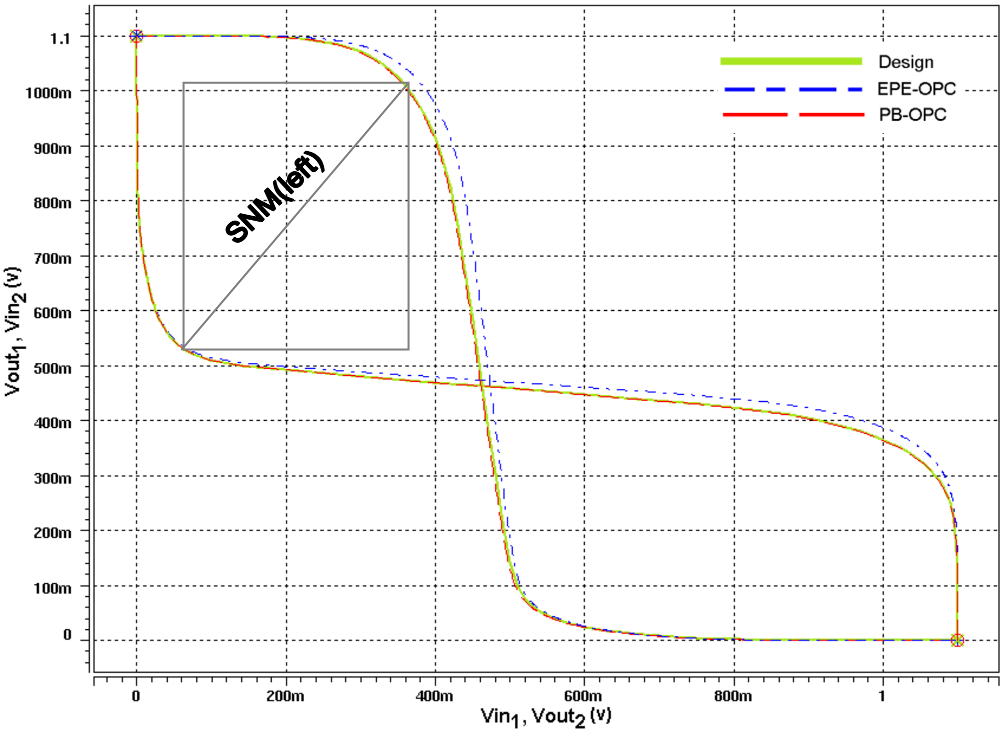


Figure 2.12: Butterfly plots of the original design, EPE-OPC and PB-OPC.

2.4 Chapter Summary

The proposed PB-OPC framework generates simpler mask that gives rise to closely matched circuit performance with the original designed value. The proposed algorithm coupled with simple mask geometry manipulation based on circuit performance reduces the mask complexity significantly. The achieved time saving in mask writing and less complicated inspection process could help offset the longer OPC run time. In addition, the run time efficiency of the PB-OPC approach is currently limited by the level of software integration and the interaction of various commercial software tools. Higher efficiency can be achieved if the algorithm is integrated into single software platform similar to the EPE-OPC approach. The methodology described in this paper is based on the ability to simulate the printed patterns and estimate these printed device's current characteristic. Hence, the accuracy of the lithography process simulation and the associated device modeling approximation are crucial in validating the proposed methodology.

Chapter 3

Device Performance-based OPC (DPB-OPC) Methodology

3.1 Introduction

The feasibility of PB-OPC has been demonstrated in Chapter 2. This chapter presents a generalized DPB-OPC with a few improvements. By simulation, DPB-OPC yields an average of 3.07% reduction in mean *Ion* deviation when compared to PB-OPC. Besides, the proposed DPB-OPC outperforms the performance-optimized EPE-OPC approach in two aspects. There is an average of 34% reduction in mask size and up to 13.5% reduction in device drive current deviation.

The rest of Chapter 3 is organized as follows. Section 3.2 describes the proposed DPB-OPC framework, with particular emphasis on the device performance extraction and mask design algorithm. Section 3.3 compares the simulation results between DPB-OPC and EPE-OPC. Finally, Section 3.4 summarizes Chapter 3.

3.2 DPB-OPC Methodology

Figure 3.1 shows the generalized DPB-OPC flow. *PDE* in (3.1) measures the deviation of the performance index of interest *P* and serves as the mask quality metric for the OPC mask design algorithm.

$$PDE = \frac{P_{post-lithography} - P_{design}}{P_{design}} \times 100\% \qquad (3.1)$$

P has to be appropriately defined for individual mask layer. For the gate region involving polysilicon and diffusion layers, the device drive current *Ion* can be a suitable performance index. For interconnect metal layers, the suitable performance index could be the interconnect delay, parasitic capacitance or resistance. Once the proper performance index is chosen, the mask design algorithm can then be formulated to ensure convergence of the correction loop under all circumstances. The complexity of the mask correction strategy can also be varied with the required precision or accuracy of performance matching. This will provide another degree of freedom in controlling the mask complexity or cost.

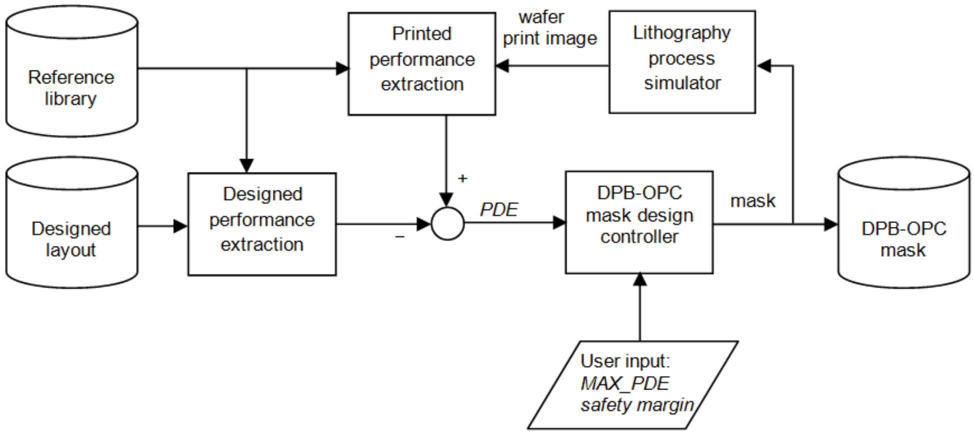

Reference library
Printed performance extraction
wafer print image
Lithography process simulator
Designed layout
Designed performance extraction
+
–
PDE
DPB-OPC mask design controller
mask
DPB-OPC mask
User input:
MAX_PDE
safety margin

Figure 3.1: The proposed DPB-OPC flow.

The DPB-OPC flow (Figure 3.1) operates as follows. Given the provided designed layout and reference library as input, the desired performance index P_{design} for each layout transistor is first extracted. The reference library contains the necessary information to aid the performance extraction process. The mask is then initialized to an exact replica of the given designed layout and subjected to the lithography process simulator to generate the resulting on-wafer printed patterns. Based on these printed patterns, the performance index $P_{post\text{-}lithography}$ will then be extracted accordingly and fed back to the mask design controller in the form of computed *PDE*. Guided by the *PDE*, the mask is corrected until *PDE* converges to the specified maximum *PDE (MAX_PDE)* and satisfies the required safety margin. The final mask is the PB-OPC mask that gives closely matched circuit performance with the designed value.

In this chapter, DPB-OPC flow is applied to design performance-based masks for both polysilicon and diffusion layers. These layers are performance-critical because they dictate the printed device shape and thus affect the final device performance. On the circuit level, the distorted printed device performance across the chip impact the overall circuit performance. Standard digital logic gates are usually characterized by their transient response, such as rise time, fall time and propagation delay. Analog circuit blocks have wider range of key performance characteristics, such as gain, power consumption, noise and bandwidth. Instead of defining different performance index for these circuits, device drive current *Ion* is chosen as the generic performance index of interest *P* for all CMOS circuits. It is chosen because it is the key parameter affecting all other circuit performances. Although it might not be the only defining parameter, simulation has shown that the proposed framework with minimizing *PDE* based on *Ion* will generally lead to other closely match circuit performance, such as the transient response (Section 3.3.2).

The performance extraction model and implemented mask design algorithm will be described in Section 3.2.1 and 3.2.2 respectively.

3.2.1 Performance Extraction Model

The gate-slicing model discussed in Section 2.2.4 is employed to extract the printed transistor performance. Note that the model assumes uniform current density along the device width direction. However, the detailed TCAD simulation revealed that the edge effect could results in a much different current density near the gate edges [54]. In addition, the threshold voltage variation over the transistor width could also affect the estimated *Ion* accuracy.

To improve the modeling accuracy, one can adopt the more accurate NRG models [54, 55]. Alternatively, one can also augment a weighting function $\gamma_k(w)$ to the existing gate slicing model (3.2), where w is the width of the sliced transistor. The weighting function $\gamma_k(w)$ can then be fitted to better match the TCAD simulation results. The simpler device characteristic extraction approach is adopted here to illustrate the potential of the proposed framework. It should be pointed out that the performance extraction is implemented as a modular block within the proposed framework, and the other more accurate NRG models can be easily integrated into the flow to offer different level of accuracy and computational speed.

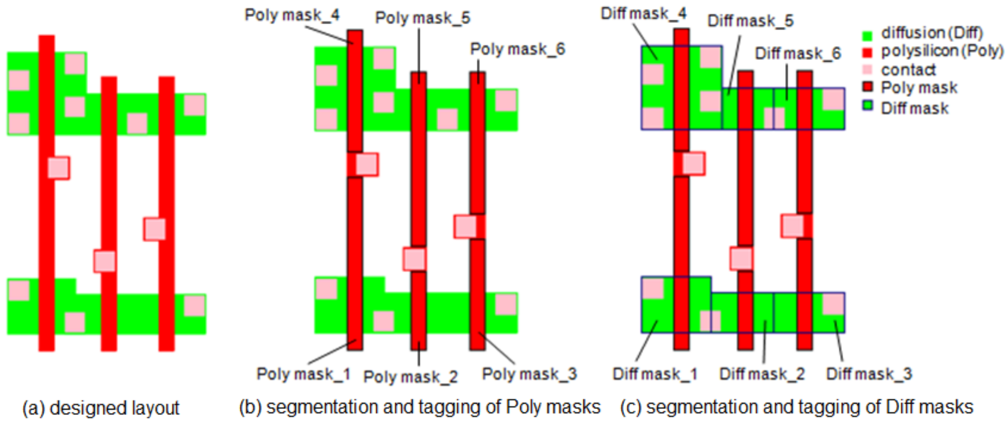
$$I_{total} = \sum_{k=1}^{m} \gamma_k(w) I_k = \sum_{k=1}^{m} \gamma_k(w) I_{library}(L_k) \times W_k \qquad (3.2)$$

3.2.2 DPB-OPC Mask Design Algorithm

The segmentation scheme described in Section *(1)* is developed to automatically tag the masks for transistors with vertical and horizontal gates. Next, the iterative mask design algorithm is described in the subsequent Section *(2)*.

(1) Segmentation Scheme

Figure 3.2 illustrates the segmentation process. Given a designed layout (Figure 3.2 (a)), the segmentation algorithm would first identify and tag the polysilicon and diffusion mask polygons for every transistor. As shown in Figure 3.2 (b), when a polysilicon (Poly) polygon is shared between multiple transistors, the algorithm would dissect it into 3 parts: Poly mask for upper transistor, Poly mask for bottom transistor, and Poly mask for connectivity. Similarly, the algorithm would also divide the shared diffusion (Diff) polygon into 3 portions: Diff mask_left transistor, Diff mask_middle transistor, and Diff mask_right transistor (Figure 3.2 (c)). Overall, this would allow independent mask adjustment between the left and the right transistor without introducing too many edge fragments.



(a) designed layout (b) segmentation and tagging of Poly masks (c) segmentation and tagging of Diff masks

Figure 3.2: Graphical illustration of segmentation process.

(2) Mask Correction Flow

Figure 3.3 displays the flowchart of the mask design algorithm. *Mask_ j(k)* is used to denote the polysilicon and diffusion mask of transistor *j* at iteration *k*.

At the first iteration, the mask (exact replica of designed layout) is subjected to lithography simulation. Based on the on-wafer printed pattern, the printed device performance and *PDE* for every transistor are extracted. Then, the *PDE* is checked against the user assigned *MAX_PDE* to decide if a correction is indeed needed. If *PDE* exceeds the provided *MAX_PDE*, a resize magnitude of *init_adjust* ΔL would be first applied to modify the respective polysilicon mask in *L* direction. The *init_adjust* value is searched from a pre-generated look-up table as a function of gate length *L* and width *W*. The characterization of *init_adjust* will be discussed later.

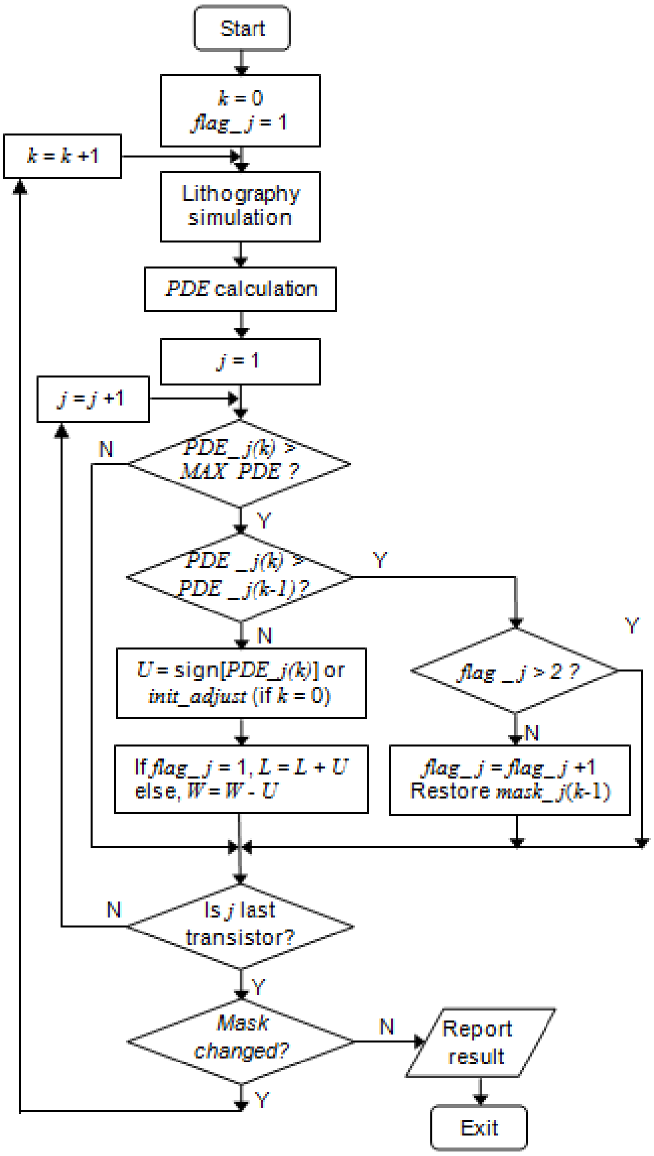


Figure 3.3: Flowchart of the DPB-OPC mask design algorithm.

After updating the mask with all necessary correction, the corrected mask is again subjected to lithography simulation and *PDE* evaluation. For subsequent iterations, the individual *PDE* will be compared to the specified *MAX_PDE*. If the specification is met, no correction will be attempted for that particular transistor's mask. Otherwise, the transistor's *PDE* trend over iterations will be analyzed to ensure that the subsequent correction leads to the minimization of its *PDE*. This is achieved by reversing the mask modification by 1nm for increasing *PDE* and continuing the mask modification by 1nm for decreasing *PDE*. The iterations stop when the specified *MAX_PDE* is met. It should be pointed out that *init_adjust* is only used for the first iteration to speed up the algorithm, and subsequent mask adjustment in either *L* or *W* direction is with a step size of 1nm.

The mask correction in *L* or *W* directions is achieved by stretching or compressing the segmented polygons in specific direction. The mask changes in *L* direction and *W* direction are regarded as coarse and fine tuning respectively due to their effects on current characteristic. Figure 3.4 shows the current characteristic plot of a NMOS transistor. *Ion* is found to be linearly proportional to *W* but inversely proportional to *L*. For a designed transistor with *L* of 65nm and *W* of 1000nm, the targeted *Ion* is 1.152mA. When the printed gate region results in larger *Ion* and thus positive *PDE*, we can either increase *L* or decrease *W* to reduce the current for the printed gate region during the next iteration. The underlying assumption is that the printed gate region will get affected similarly in terms of its effective gate length and effective gate width. In the proposed mask design algorithm, *L* is modified first to provide coarse tuning due to its larger effect on *Ion*. Fine tuning will be achieved by modifying *W* subsequently due to its smaller impact on *Ion*.

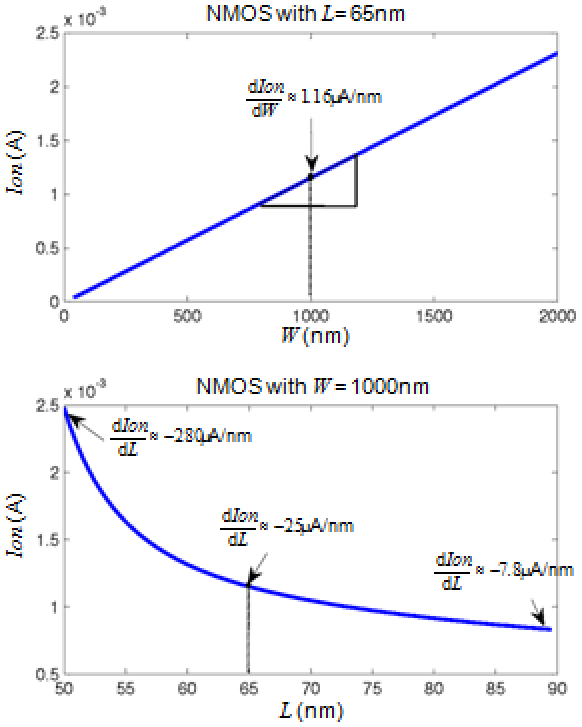


Figure 3.4: Dependency of current *Ion* on *W* and *L*.

To prevent catastrophic open failure of diffusion layer, adjustment in *L* direction will only be performed on polysilicon mask. This is to prevent the adjacent diffusion mask polygons (e.g. Diff mask for transistor 1, 2 and 3 in Figure 3.2) become disconnected after shrinkage operation in *L* direction. Similarly, adjustment in *W* direction will only be performed on diffusion mask. On the other hand, the *L* adjustment for polysilicon mask might potentially lead to the bridging of neighboring devices and reduce the transistor counts. Therefore, a checking algorithm is included in the mask design algorithm during *PDE* estimation to monitor the transistor counts. Once the violation is detected, the current *L* adjustment would be reversed and the algorithm would proceed with *W* adjustment to minimize the *PDE* for the affected transistors. The bridging in *W* adjustment is not an issue in the proposed framework due to the large spacing between the neighboring diffusion polygons imposed by the design rule.

In summary, the correction step is performed for every transistor based on the individual *PDE* rating independently. It is performed iteratively until the individual transistor's *PDE* fall within the user-assigned tolerance *MAX_PDE*. The proposed algorithm ensures local minima solution by two conditions:

- The starting point of search is initialized to the vicinity of the minima point.
- Local iterative gradient descent method is used to minimize the *PDE*.

The *init_adjust* is used during the first iteration to speed up the mask design algorithm. The characterization of *init_adjust* is described as follows. First, an evaluation transistor (Figure 3.5) is proposed to characterize the optimum *init_adjust* for first iteration. The value of polysilicon line-end extension past active of 84nm and active enclosure of gate of 98nm are chosen according to the 65nm design rule. Next, the evaluation transistor with *L* of 70nm and *W* of 100nm is subjected to DPB-OPC flow with *init_adjust* prefixed to ΔL. The ΔL value is varied from -4nm to 15nm with step size of 1nm. For each ΔL value, the final performance *PDE* is evaluated. The ΔL value that results in minimum *PDE* is chosen as *init_adjust* for *L* of 70nm and *W* of 100nm. The *init_adjust* characterization process is repeated for other combinations of *L* and *W.* A look-up table is constructed to map the *init_adjust* value as a function of gate length *L* and width *W.* Therefore, during the first iteration in the proposed mask design algorithm, *init_adjust* can be chosen depending on the *L* and *W* to bring the *PDE* closer to the specified *MAX_PDE*. This will reduce the number of iterations required compared to the case where a uniform step size of 1nm is chosen for all iterations.

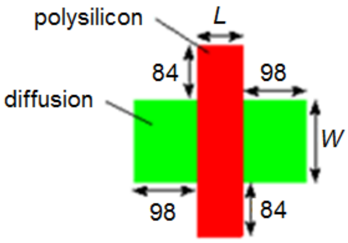


Figure 3.5: Model for characterizing *init_adjust* (dimension is in nanometer).

Chapter 2 [65] proposed a rather different mask design algorithm, in which the *init_adjust* is determined from the difference in effective gate length, i.e. $init\_adjust = L_{\text{design}} - Leq_{Ion}$. In addition, it only handles layout of vertical gates. In this chapter, the proposed DPB-OPC algorithm can handle layout with both horizontal and vertical gates. The minimization of performance deviation error is also further improved through a systematic approach. Figure 3.6 shows that the performance is improved by an average of 3.07% reduction in mean *Ion* deviation compared to [65].

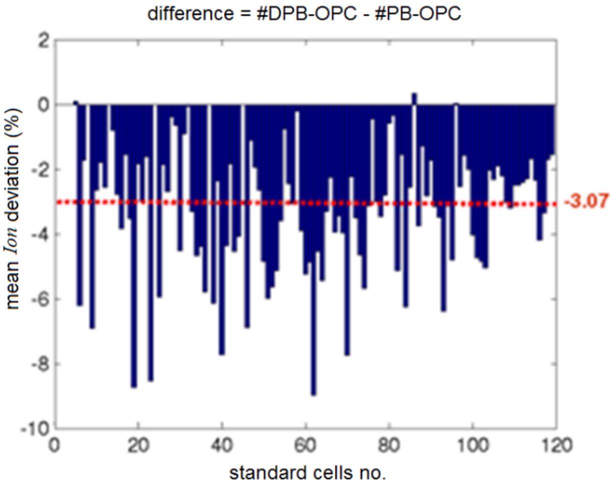
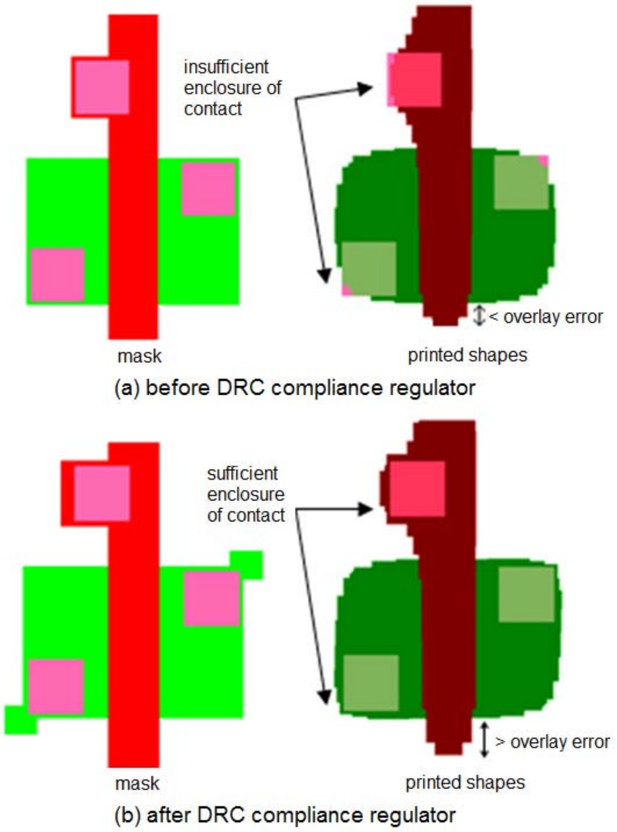


Figure 3.6: Performance difference between DPB-OPC and PB-OPC [65].

(3) DRC compliance regulator

To prevent catastrophic failures, it is important to ensure that the post DPB-OPC layout do not exhibits bridging, pinching, open or short issues even in the presence of mask misalignment. As far as the diffusion and polysilicon layers are concerned, the relevant failure mechanisms are bridging between transistors, bridging between polysilicon to neighboring contacts, line-end pull back with overlay errors and insufficient enclosure of contact.

The detection and elimination of bridging of transistors has been mentioned in earlier Section 3.2.2*(b)* through the monitoring of the transistor counts. Similar concept can be applied to the bridging between polysilicon to neighboring contacts. It is worth mentioning that due to the sufficient spacing between the contacts and polysilicon stipulated by DRC rules, we have never encountered the latter bridging throughout the simulation. To avoid line-end pull back with overlay errors, the resulting polysilicon to diffusion extension for printed shapes will be estimated to ensure minimum extension margin (>overlay errors). Larger polysilicon to diffusion extension would be given to the designed mask if necessary as shown in Figure 3.7. Lastly, to ensure sufficient enclosure of contacts, larger enclosure for both polysilicon and diffusion layers can be applied as shown in Figure 3.7. All the above-mentioned techniques have been implemented in the proposed algorithm through a modular block called DRC compliance regulator.



(a) before DRC compliance regulator

(b) after DRC compliance regulator

Figure 3.7: Before and after the DRC compliance regulator.

3.3 Results and Discussions

The proposed DPB-OPC framework is implemented using Perl script. To verify the effectiveness of the proposed DPB-OPC methodology, the 65nm standard cells library and IEEE International Symposium on Circuits and Systems 1985 (ISCAS’ 85) benchmark circuits are used as the test vehicles. Based on the provided 65nm library, the ISCAS’ 85 benchmark Verilog circuits from [66] are first synthesized using Synopsys Design Compiler version W2004.12-SP4 [67]. The synthesized netlists are then placed and routed using Cadence SOC_Encounter 7.1 [68]. The generated layouts are then fractured to MEBES format employing Calibre Workbench. The resulting MEBES file, which is a common format for raster scan mask writing machine, can serve as an indicator for the mask complexity. Table I shows the transistor count as well as the original designed mask MEBES size for diffusion and polysilicon layers.

Table 3.1: Benchmark circuit specification.

Circuit	# Transistor	Diffusion mask size (Bytes)	Polysilicon mask size (Bytes)
c432	662	12288	30720
c499	1862	24576	63488
c880	1778	24576	65536
c1908	2032	24576	69632
c2670	4934	69632	178176
c3540	6250	79872	212992
c5315	10738	135168	370688
c6288	8202	75776	294912
c7552	12586	161792	438272

3.3.1 Performance-optimized EPE-OPC Mask Generation

To facilitate comparison between EPE-OPC and DPB-OPC, Calibre OPCpro [69] is employed to generate the edge placement error based OPC (EPE-OPC) mask. The Calibre OPCpro corrects the edge placement errors by moving individual fragments at control sites based on simulated EPE data using computer models of the optical system and the lithographic processes. Fragments will be moved iteratively with step size adjusted according to the EPE value. It should be noted that different OPCpro configuration will result in different EPE-OPC mask and circuit performance. Figure 3.8 shows an example of how performance (mean *Ion* deviation) varies with OPCpro settings. Here, the parameters minedgelength, concavecorn, cornedge, and ripplelen [69] are grouped under single term called fragmentation length. These settings define the fragmentation scheme employed during OPC optimization. In general, large fragmentation length results in low fragmentation and thus less complex mask.

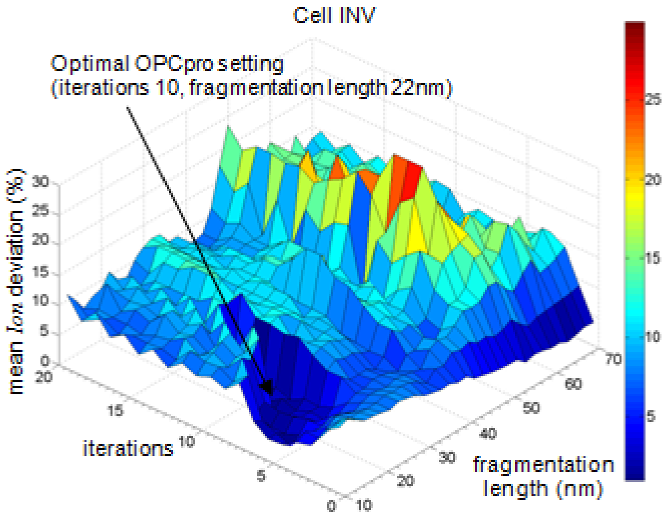


Figure 3.8: Mean *Ion* deviation varies with OPCpro setting.

In order to obtain the optimal OPCpro settings that yield optimal device performance (which is minimal mean *Ion* deviation in this case), a rigorous search from the defined search space is performed. The search space is bounded by fragmentation length of 10nm to 70nm in step of 2nm, iterations of 1 to 20 in step of 1, and step size of 1nm which constitutes 620 possible combinations. Among these combinations, the OPCpro settings that results in minimum mean *Ion* deviation will be used to generate the EPE-OPC mask for comparison with the proposed DPB-OPC. We conduct the search for every test case such that the "benchmark" EPE-OPC masks' performance are optimal to impose stiff competition for subsequent comparison with the DPB-OPC masks.

3.3.2 Comparison with EPE-OPC Methodology

Both DPB-OPC and EPE-OPC approaches are compared in 3 aspects, i.e. drive current deviation, mask size, and run time. The mask is subjected to lithography simulation such that the performance deviation can be determined from the post-lithography printed patterns. The chosen performance indices are defined as follows:

Ion deviation for transistor with index j,

$$d_j = \left| \frac{Ion_{post-lithography,j} - Ion_{design,j}}{Ion_{design,j}} \right| \times 100\% \quad , \qquad (3.3)$$

mean *Ion* deviation,

$$\bar{d} = \frac{\sum d_j}{N} \quad , \qquad (3.4)$$

standard deviation (stdv) of *Ion* deviation,

$$\sigma_d = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(d_j - \bar{d}\right)^2} = \sqrt{\frac{1}{N}\left(\sum_{j=1}^{N} {d_j}^2\right) - \left(\bar{d}\right)^2} \quad , \qquad (3.5)$$

maximum *Ion* deviation,

$$d_{max} = \max_{\forall j} d_j \quad , \qquad (3.6)$$

and minimum *Ion* deviation,

$$d_{min} = \min_{\forall j} d_j \quad . \qquad (3.7)$$

These indices could provide a direct measure of the OPC correction quality as well as a rough estimate for the overall circuit performance without explicit circuit simulation on the post-OPC netlist. In addition, the equivalent channel length (*Leq_{Ion}*) is also extracted based on the total NRG drive current and is used for transient response analysis [53]. To verify the correlation between *Ion* and the other performance characteristics, detailed post-OPC circuit simulation is performed on the 94 standard cells. For all input transition scenarios, the transient performance in terms of propagation delay t_p, rise time t_r, and fall time t_f are extracted. Only the worst case transient performance deviations are used for comparison and are defined as follows:

$$\text{worst case } t_p \text{ deviation} = \max_{\forall j}\left(\left|\frac{t_{p\, post-lithography,j} - t_{p\, design,j}}{t_{p\, design,j}}\right| \times 100\%\right), \qquad (3.8)$$

$$\text{worst case } t_r \text{ deviation} = \max_{\forall j}\left(\left|\frac{t_{r\, post-lithography,j} - t_{r\, design,j}}{t_{r\, design,j}}\right| \times 100\%\right), \qquad (3.9)$$

$$\text{worst case } t_f \text{ deviation} = \max_{\forall j}\left(\left|\frac{t_{f\, post-lithography,j} - t_{f\, design,j}}{t_{f\, design,j}}\right| \times 100\%\right). \qquad (3.10)$$

Figure 3.9 illustrates the correlation between the mean *Ion* deviation and the worst case $t_p/t_r/t_f$ deviation for the 94 sampled digital standard cells. In general, the distribution of mean *Ion* deviation roughly correlates with the trend line of transient performance deviation. This justifies the choice of *Ion* as the chosen performance index for the proposed DPB-OPC.

As the OPC mask size in MEBES format serves as a good indicator for the mask complexity and thus the mask cost, it is used for cost comparison between the two approaches. Finally, the DPB-OPC run time to the EPE-OPC search time required for obtaining the performance-optimized EPE-OPC mask is also compared.

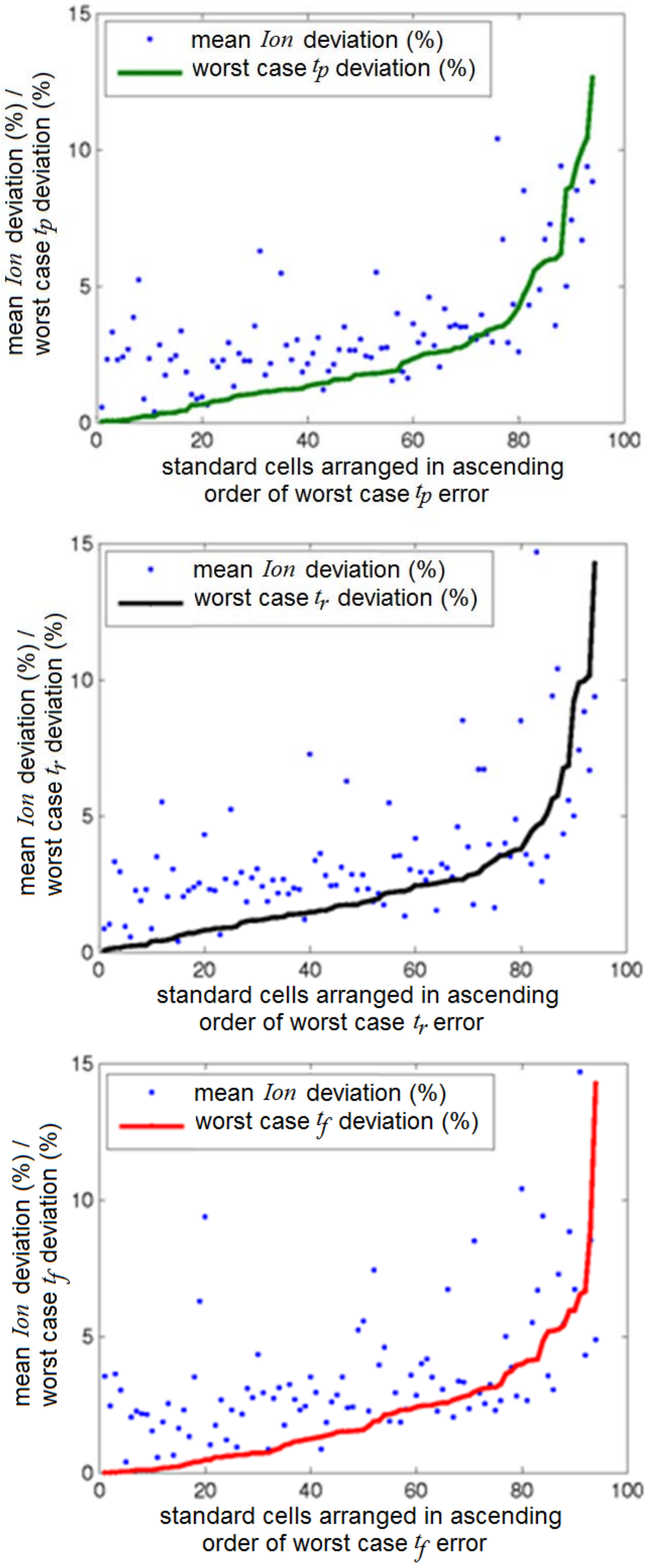


Figure 3.9: Correlation of transient performance with mean *Ion* deviation error.

(1) Standard cells library

Figure 3.10 and Figure 3.11 summarize the comparison of circuit performance and mask size for all 119 standard cells. Figure 3.10 plots the performance difference by subtracting the performance index value of EPE-OPC from DPB-OPC. A negative value of x % implies that the DPB-OPC reduce the performance index by x % and is therefore desired.

Among the 119 test cases, 96.6% or 115 achieve reduction in mean *Ion* deviation. An average of 80% of the test cases was also observed to achieve reduction for the remaining performance indices such as standard deviation, maximum, and minimum of *Ion* deviation distribution. Considering all 119 test cases, the average reduction in *Ion* deviation is 2.42% for mean, 2.24% for standard deviation, 8.56% for maximum, and 0.3% for minimum.

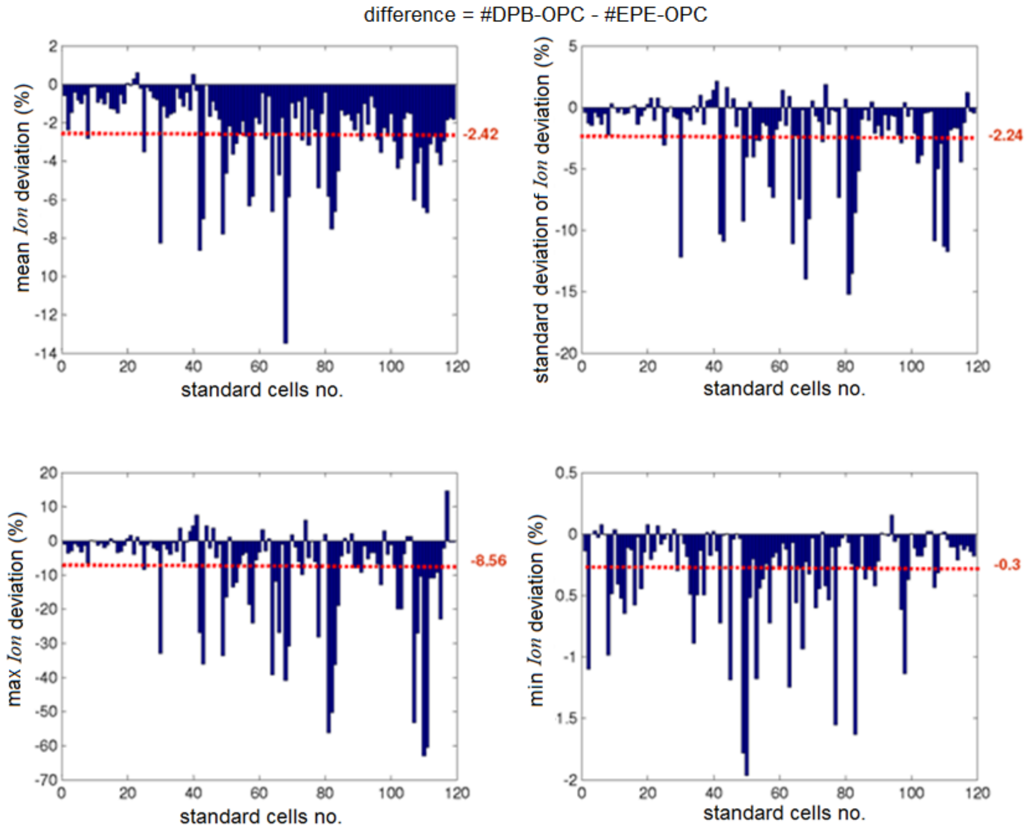


Figure 3.10: Performance difference between DPB-OPC and EPE-OPC.

On the other hand, Figure 3.11 reveals that the proposed DPB-OPC methodology achieves an average of 33.3% mask size reduction across all test cases. Reduced mask size translates to reduced mask writing time as well as less complicated fabrication and inspection process, which implies reduced mask cost. It should be pointed out that only diffusion and polysilicon layers are considered for the mask size comparison. The reported reduction would be smaller if non-transistor regions and other mask layers are considered which might require more fragmentations for the proposed OPC.

Run time and iterations required by DPB-OPC approach are reported in Figure 3.12. When compared to the search time required for obtain the performance-optimized EPE-OPC mask, over 77% of time saving is achieved.

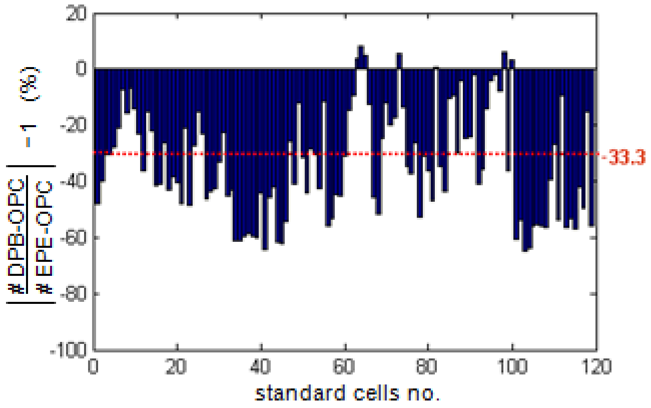


Figure 3.11: Mask size comparison (diffusion and polysilicon masks) between DPB-OPC and EPE-OPC.

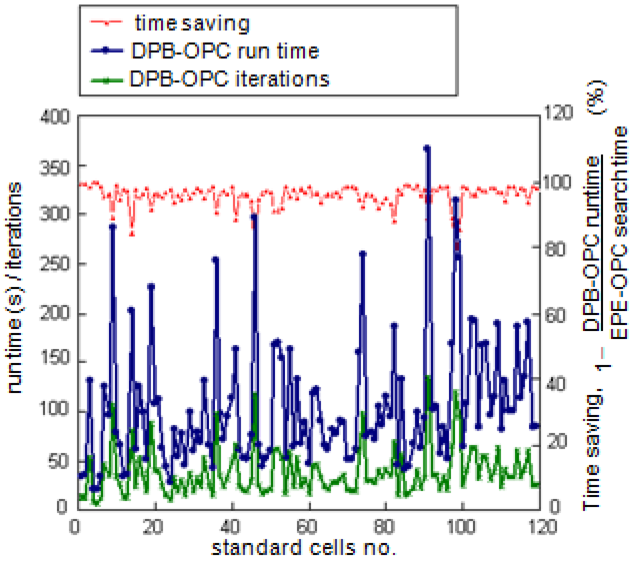


Figure 3.12: Run time comparison between DPB-OPC and EPE-OPC.

(2) ISCAS' 85 Benchmark Circuits

The ISCAS' 85 benchmark circuits are subjected to both DPB-OPC and EPE-OPC methodology to facilitate comparison.

Table 3.2 to Table 3.4 summarize the simulation results and the comparison statistic. The DPB-OPC outperforms the performance-optimized EPE-OPC and achieves 1.7% to 3.7% reduction in mean *Ion* deviation, 34% average reduction in mask sizes, and at least 58.6% run time saving. As shown in Table 3.2, the maximum *Ion* deviation and its sigma (the spreadness) are also improved by minimum 27.9% and 1.9% respectively. As for Table 3.3, the mask size reduction is seen across all circuits of various size and polysilicon masks dominate the saving. Table 3.4 shows that the OPC run time generally tracks with the transistor counts. Figure 3.13 shows the resulting simpler DPB-OPC mask and the performance-optimized EPE=OPC mask of circuit c432.

Table 3.2: Comparison of post-OPC circuit performance.

Circuit	EPE-OPC *Ion* deviation (%)				DPB-OPC *Ion* deviation (%)				Difference, DPB−EPE (%)			
	Mean	Max	Min	Stdv	Mean	Max	Min	Stdv	Mean	Max	Min	Stdv
c432	5.2	72.6	0.00652	7.9	1.4	12.3	0.00011	2.6	-3.7	-60.3	-0.00641	-5.4
c499	5.5	90.0	0.00107	9.4	1.8	10.4	0.00018	2.6	-3.7	-79.6	-0.00089	-6.8
c880	3.0	59.1	0.01837	5.5	1.2	12.3	0.00117	2.2	-1.7	-46.8	-0.01720	-3.3
c1908	4.4	59.1	0.00326	6.4	1.6	12.3	0.00007	2.6	-2.8	-46.8	-0.00320	-3.8
c2670	2.8	48.5	0.01837	3.6	0.8	9.4	0.00117	1.6	-2.0	-39.1	-0.01720	-2.0
c3540	3.3	59.1	0.00440	4.2	1.1	12.3	0.00041	2.0	-2.1	-46.8	-0.00399	-2.2
c5315	3.0	39.2	0.00215	3.7	0.9	11.3	0.00070	1.8	-2.1	-27.9	-0.00145	-1.9
c6288	4.0	81.3	0.01262	4.3	1.2	12.3	0.00007	2.4	-2.7	-69.0	-0.01255	-1.9
c7552	3.0	39.2	0.00440	4.3	0.9	10.4	0.00070	1.7	-2.1	-28.8	-0.00369	-2.6

Table 3.3: Comparison of mask size.

Circuit	EPE-OPC mask size (Bytes)		DPB-OPC mask size (Bytes)		Improvement, |DPB/EPE| − 1 (%)	
	Diffusion	Polysilicon	Diffusion	Polysilicon	Diffusion	Polysilicon
c432	28672	63488	26624	30720	-7.1	-51.6
c499	71680	196608	57344	75776	-20.0	-61.5
c880	69632	147456	61440	73728	-11.8	-50.0
c1908	65536	174080	59392	79872	-9.4	-54.1
c2670	194560	387072	172032	202752	-11.6	-47.6
c3540	249856	477184	200704	239616	-19.7	-49.8
c5315	407552	823296	346112	432128	-15.1	-47.5
c6288	249856	942080	190464	335872	-23.8	-64.3
c7552	493568	966656	397312	505856	-19.5	-47.7

Table 3.4: Comparison of OPC run time.

Circuit	EPE-OPC	DPB-OPC		Time saving, 1- |DPB-OPC run time /EPE-OPC search time| (%)
	Search time (s)	Run time (s)	Iterations	
c432	7477.8	622.1	118	91.7
c499	8410.6	611.3	69	92.7
c880	1576.3	653.4	86	58.6
c1908	27844.5	1518.9	118	94.5
c2670	62895.7	3194.7	86	94.9
c3540	88524.2	5724.2	118	93.5
c5315	106619.1	9828.8	118	90.8
c6288	100851.3	4901.9	88	95.1
c7552	124001.8	11639.4	118	90.6

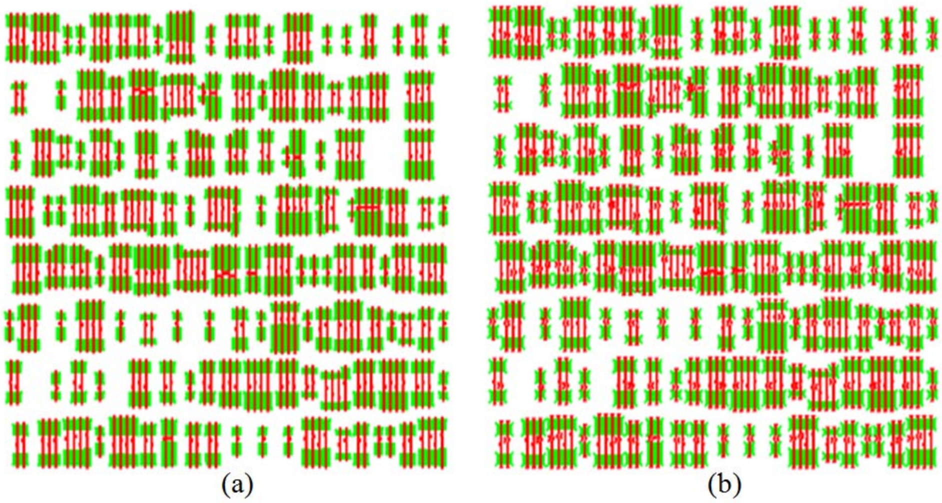


Figure 3.13: Comparison between (a) DPB-OPC mask and (b) EPE-OPC mask of circuit c432.

3.3.3 Investigation of Post-OPC Path Delay

For the path delay comparison, the post DPB-OPC and post EPE-OPC netlists are generated for SPICE transient simulation. The netlist is obtained by replacing the transistor *L* and *W* with the Leq_{Ion} and Weq_{Ion}, where Leq_{Ion} and Weq_{Ion} are the equivalent channel length and width extracted based on the total NRG drive current for both OPCs [53].

Reference [53] suggests the modeling of the load capacitance through modifying the *DLC* parameter for each transistor. The *DLC* parameter is calculated as follows:

$$DLC = LINT + \frac{Leq_{Ion} - Lave}{2}, \tag{3.11}$$

$$Lave = \frac{NRG\_gate\_area}{Weq_{Ion}}, \tag{3.12}$$

where *LINT* is empirical BSIM parameter.

To model the intrinsic capacitance, fringe capacitance and overlap capacitance, the BSIM 4 model [53] defines L_{active} and W_{active} as follows:

$$L_{active} = Leq_{Ion} + XL - 2DLC = Lave + XL - 2LINT \,, \tag{3.13}$$

$$W_{active} = Weq_{Ion} - 2WINT \,, \tag{3.14}$$

where *XL* is channel length offset due to mask or etch effect, and *WINT* is empirical BSIM parameter.

Through (3.11) to (3.14), it can be shown that the resulting gate capacitance would be related to NRG_gate_area, which is extracted based on the on-wafer printed patterns, whereas the drive current is still governed by Leq_{Ion} and Weq_{Ion}.

For a circuit having m input ports and n output ports, given four possible input transition scenarios (i.e. 0→0, 0→1, 1→0, 1→1), it will results in a total of $m^4 - m^2$ useful input transition patterns and up to a maximum $n \times (m^4 - m^2)$ possible output transitions. Any input transition that leads to output transition will give rise to a path delay (t_d), which is defined as the delay between the two transitions. A Perl script was written to automate these path delays simulation and extraction process. Based on the collected data, the t_d deviation is obtained as follows:

$$t_d \text{ deviation} = \left| \frac{t_{d\,post-OPC} - t_{d\,design}}{t_{d\,design}} \right| \times 100\% \tag{3.15}$$

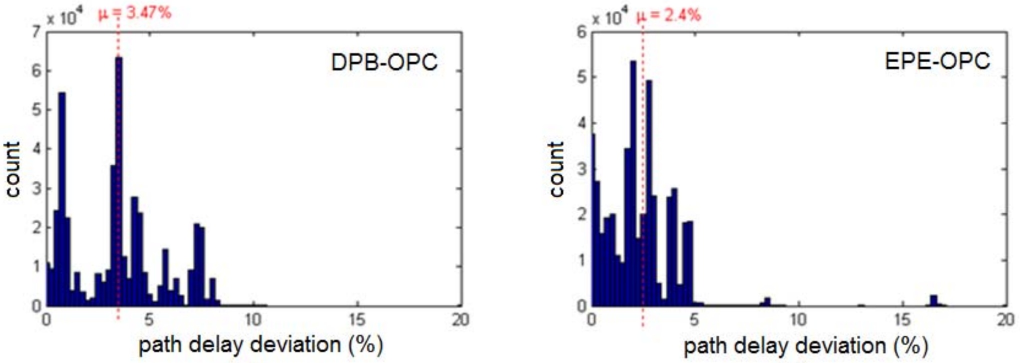
(1) c1908

Among the ISCAS85’ circuits, the medium size circuit c1908 is chosen to evaluate the post-OPC circuit performance comparison. The circuit c1908 consists of 33 inputs and 25 outputs. This results in $33^4 - 33^2$ possible input transition patterns, which cannot be fully simulated in real time. In this comparison, only 2% or 22000 input transition patterns are randomly selected for SPICE simulation to extract the path delay. The resulting histogram of path delay deviation for both post-DPB-OPC and post-EPE-OPC are displayed in Figure 3.14. The mean path delay deviation for DPB-OPC and EPE-OPC are 3.47% and 2.4% respectively. As expected, the DPB-OPC results in slightly worse performance due to the larger gate capacitance deviation. However, the deviation is small enough to justify the choice of *Ion* as the key performance metric in the proposed algorithm. The smaller deviation in mean path delay for DPB-OPC compared to gate capacitance deviation is due to the better matching in *Ion* as well as the possibly larger transistor width employed in the synthesized circuit which leads to smaller gate capacitance variation as pointed out in Figure 3.16. Other path delay metrics are also listed in Table 3.5.

Table 3.5: Comparison of post-OPC path delay deviation.

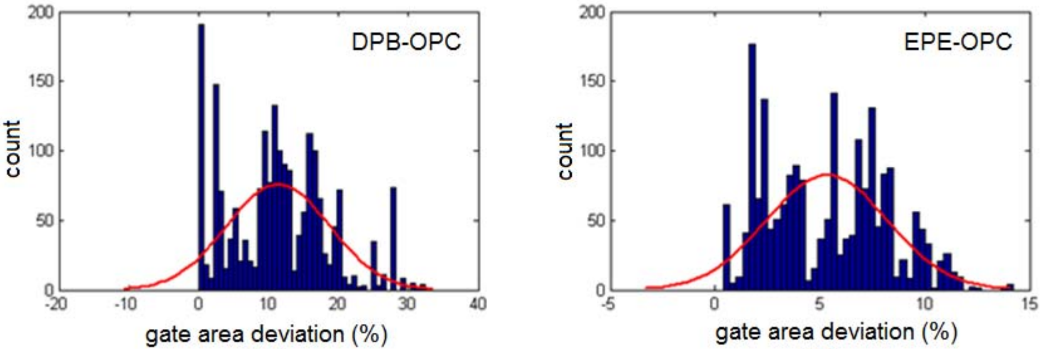
	Performance deviation (%)	
	EPE-OPC	DPB-OPC
Maximum path delay	1.03	0.39
Minimum path delay	0.37	1.87

As the proposed DPB-OPC only targets at matching the desired *Ion* rather than the desired printed patterns, the resulting gate capacitance for the transistor is expected to deviate from the designed transistor. As the gate capacitance is directly related to the gate area, the capacitance deviation can be gauged by examining the resulting NRG gate area from the on-wafer printed patterns. As shown in Figure 3.15, the mean NRG gate area deviation for DPB-OPC and EPE-OPC are 11.37% and 5.34% respectively. Although the DPB-OPC is expected to have larger capacitance deviation, the error is within process variation. In addition, it is also observed that the area deviation becomes comparable for both OPCs when the transistor width gets larger (Figure 3.16).



(a) (b)

Figure 3.14: Histogram of path delay deviation for (a) post DPB-OPC c1908 and (b) post EPE-OPC c1908.



(a) (b)

Figure 3.15: Histogram of gate area deviation for (a) post DPB-OPC c1908 and (b) post EPE-OPC c1908.

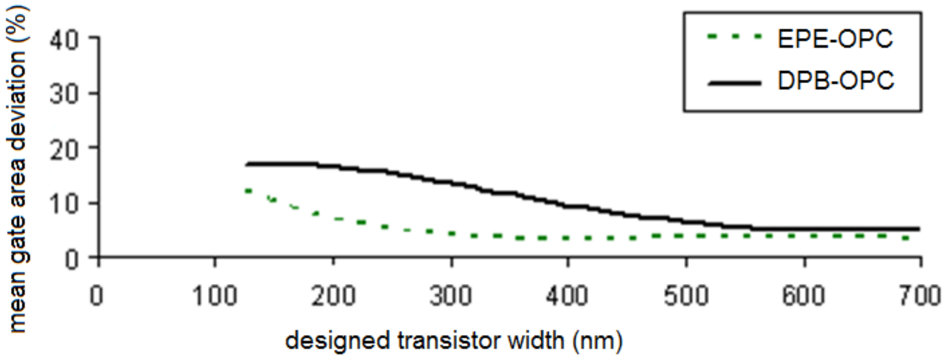


Figure 3.16: Trendline relationship between the mean gate area deviation and the designed transistor width.

3.4 Chapter Summary

This chapter proposes a device performance-based OPC framework to design mask that give rise to closely matched circuit performance with the designed value. The proposed algorithm coupled with simple mask geometry manipulation reduces the mask complexity. In comparison to the conventional EPE-OPC approach, the performance-aware and mask cost-aware DPB-OPC approach achieves an average of 34% mask size saving, and up to 13.5% of reduction in the mean drive current deviation. It should be noted that the comparison was only performed for polysilicon and diffusion mask layers with emphasis on transistor region. By considering non-transistor region as well as other mask layers, the reported mask size reduction could be smaller. It is worth pointing out that the complexity of the mask correction strategy can also be varied with the required precision or accuracy of performance matching. This will provide another degree of freedom in controlling the mask complexity or cost. Besides, since the proposed framework is implemented in modular structures, other process simulation models and device performance extraction models can be added in easily.

Chapter 4

Library-based Device Performance-based OPC for Hierarchical Circuits

4.1 Introduction

DPB-OPC framework proposed in Chapter 3 has shows promising results in achieving considerable mask data saving as well as improved circuit performance matching. Despite this, the performance gain is currently limited by the comparatively longer run time. Due to its iterative performance evaluation of every transistor, the OPC run time is also anticipated to be exponentially increasing with the number of transistors; and therefore prohibiting its effective application on the VLSI circuit comprised of billions of transistor.

To improve run time efficiency of the proposed DPB-OPC [56], a library-based DPB-OPC methodology for VLSI digital circuit is presented in this chapter. The cell-wise OPC strategy [32] is borrowed and adapted into the framework to explore its merit of run time saving. Fundamentally, major portion of the VLSI digital chip is synthesized and composed of instantiated standard cell layouts from the provided foundry libraries. By first pre-characterizing the OPC mask for each standard cell during the library database construction, the entire full chip OPC mask can then be formed by stitching the respective cells' OPC mask per synthesized placement order and thus results in shortened computational time.

However, the non-negligible optical proximity effects (OPE) introduced by boundary cells, especially evident around the cell boundaries region, could contribute to different printing result between the cell-wise OPC and conventional model-based OPC. This in-turn results in performance disturbance to the transistors at the boundary regions.

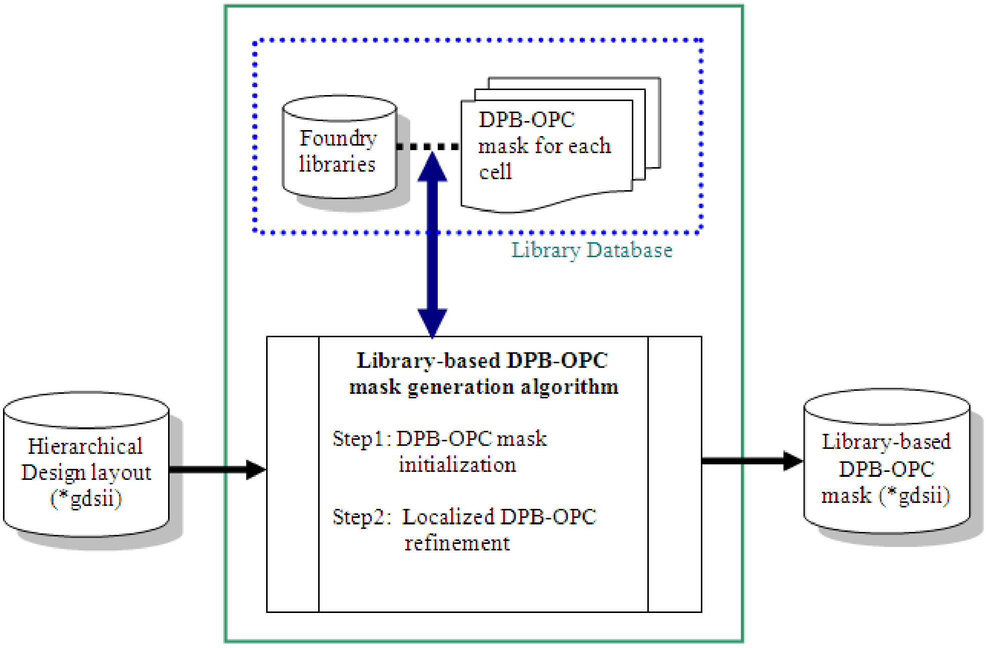
Various methods were proposed in literature to reduce the discrepancy due to OPE [31, 32, 70-72] but all are based on the aim of geometrical shape or critical dimension reproduction. The proposed cell wise OPC methodology [32] employs dummy features to represent the neighboring cells environment for polysilicon and contact layers which leads to at most 6% error when compared to full-chip OPC in 90nm design. Considering the possible insufficiency of dummy features as environmental representation at advanced technology node beyond 90nm, Wang et al. [70] proposed cell-divided core and boundary parts driven OPC. In particular, the standard cell is divided into two portions: core and boundary; then the OPC solutions for core part are pre-characterized during library construction while the left-over boundaries part accept full-chip OPC after placement. On the other hand, Kahng et al.[31] proposed auxiliary patterns (AP) based OPC that shields polysilicon patterns from proximity effects of neighboring cells, thus eliminating the localized OPC refinement and achieving better gate polysilicon EPE count matching with that of model-based OPC. A 4% inaccuracy were achieved in AP-OPC, an average of 90% improvement was seen over the cell wise OPC without AP proposed in [32]. Similarly, an environment specific boundary-based approach was proposed in [71]. With the assumption of only vertical features and fixed pitch considered, each standard cell is OPC-corrected

48 times and stored accordingly with respect to the 48 different representative environments. As a result, the corrected boundary features could be conditionally substituted based on neighboring cells in the full-chip layout. The method has reportedly achieved a 6× improvement on average EPE and 2× improvement on maximum width error over non-boundary based approach. Despite of focusing on cell wise OPC library characterization, a different hierarchical cell wise OPC [72] method is approached by segment-moving map and dynamic correction to identify the interacting regions and automatically adjust the corrections in these regions. A 5× speed up was achieved with similar accuracy when compared with the full chip OPC method. With the renewed objective of minimizing the electrical performance in DPB-OPC, the electrical impact of the OPE on cell boundaries are indeed the proper metric to be acquired and will be used to guide the dynamic correction.

This chapter is organized as follows. Section 4.2 explains the library-based DPB-OPC flow. The simulation setup and results are covered in Section 4.3. Chapter summary is provided in Section 4.4.

4.2 Library-based DPB-OPC Flow

The proposed library-based DPB-OPC methodology is shown in Figure 4.1. Given a hierarchical design layout, an initialized DPB-OPC mask for the entire layout of synthesized digital circuit is formed by stitching the pre-characterized DPB-OPC mask of the respective instantiated standard cell from the library database. However, the proximity effects induced by different surrounding environment could affects the post-lithography print image and thereby the device performance. To rectify the OPE-caused performance shift, an adaptive correction which based on the diagnosed "device performance disturbance" is then performed during the localized DPB-OPC refinement phase. Final output, which is a library-based DPB-OPC mask that results in printed patterns match closer to the designed circuit performance, is output to the user. The detailed explanations of the library database and library-based DPB-OPC are presented as follows.



Foundry libraries
DPB-OPC mask for each cell
Library Database
Library-based DPB-OPC mask generation algorithm
Step1: DPB-OPC mask initialization
Step2: Localized DPB-OPC refinement
Hierarchical Design layout (*gdsii)
Library-based DPB-OPC mask (*gdsii)

Figure 4.1: Library-based DPB-OPC methodology.

4.2.1 Library Database

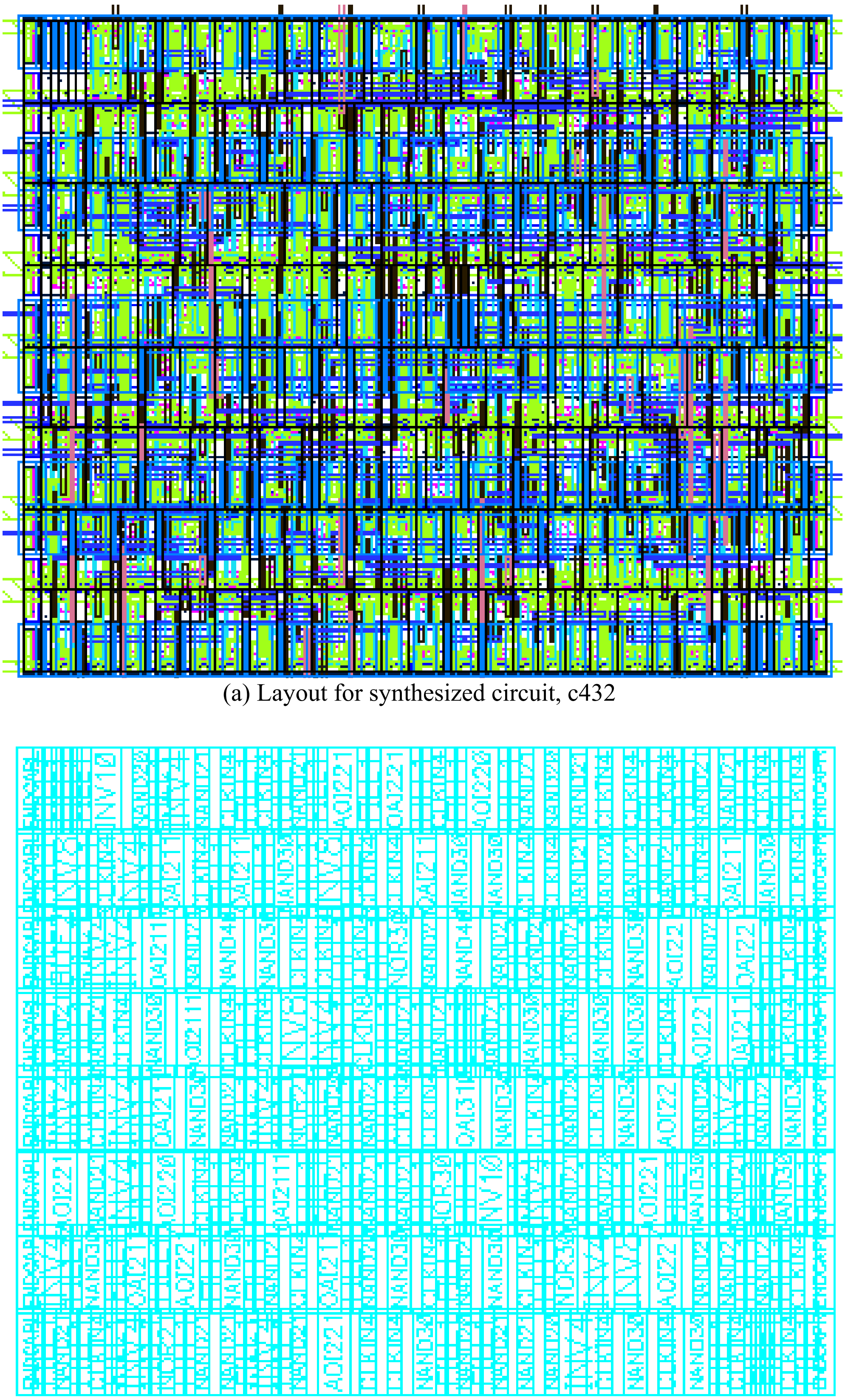
The DPB-OPC mask for standard cells layout of foundry libraries are generated using the framework presented in Chapter 3 and indexed accordingly in the library database. This process is thus referred as "standard cell wise DPB-OPC".

4.2.2 Library-based DPB-OPC Mask Generation Algorithm

The proposed algorithm consists of two steps:

- Step 1 – DPB-OPC mask initialization
- Step 2 – Localized DPB-OPC refinement

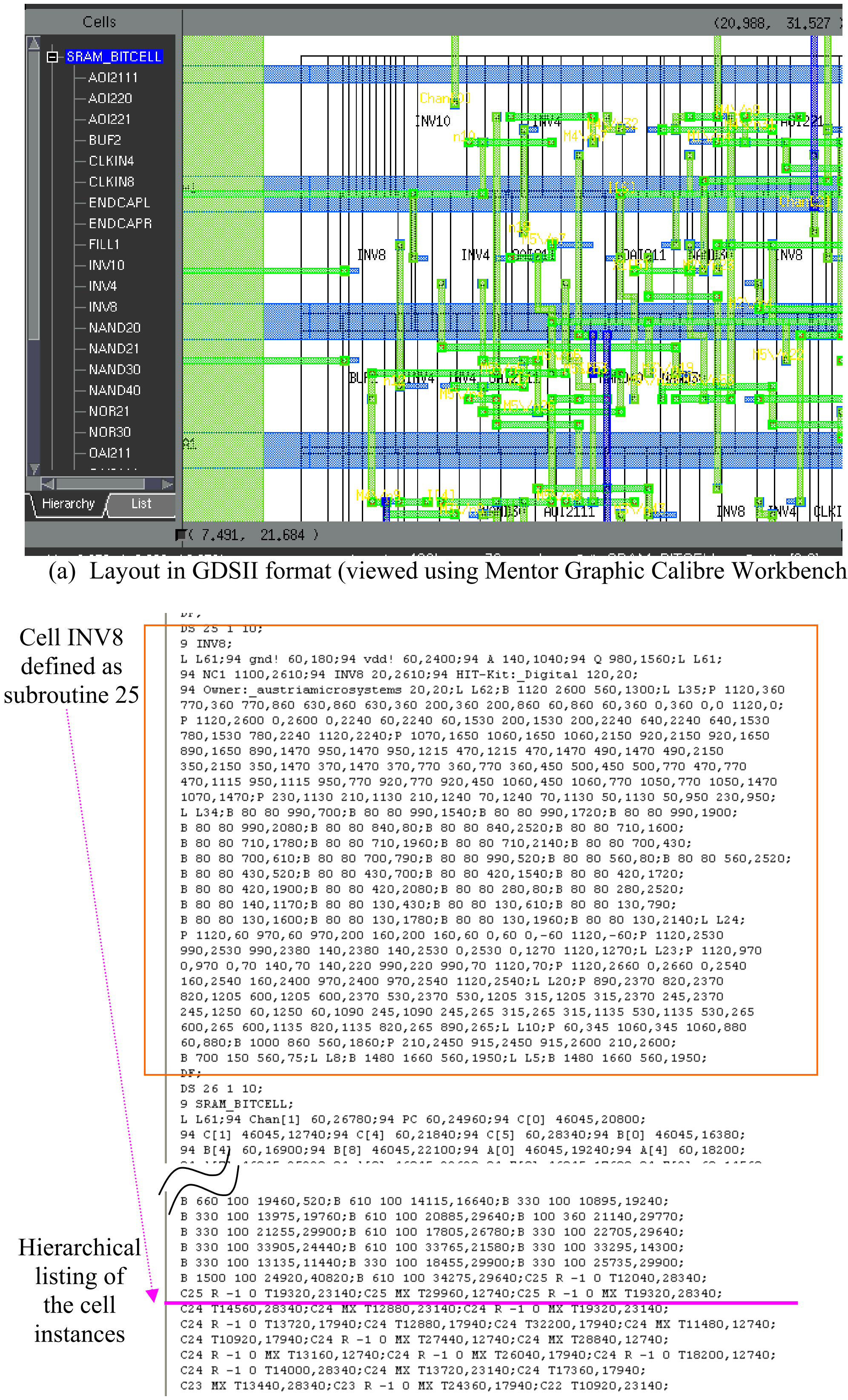
In step 1, an initialized DPB-OPC mask for the entire layout of synthesized circuit is first formed by stitching the pre-characterized DPB-OPC mask of the respective instantiated standard cell per placement order. Note that the layout of any synthesized circuit is basically the combination of many and different cell instances together. An example of layout and cell view for a synthesized circuit c432 is given in Figure 4.2. Figure 4.3 shows the hierarchical layout for c432 in the two common describing formats: binary GDSII format and in CIF text format. Figure 4.3(a) shows the GDSII layout viewed using Mentor Graphic Calibre Workbench in which the hierarchical listing is clearly displayed in the left panel. However, such hierarchical information is embedded inside the binary coded format and difficult to be extracted by normal text processing in Perl. Therefore GDSII format has to be converted into the CIF text format. Figure 4.3(b) shows the layout description in CIF text format in which the hierarchical listings of all cell instances are clearly displayed. Hence, the placement order can be easily retrieved from the hierarchical listing through text processing. Since only database look-up operation and geometry manipulation are required during this step, this result in significant saving in computational time when compared to the conventional full chip DPB-OPC.



(a) Layout for synthesized circuit, c432

(b) Cell view for synthesized circuit c432

Figure 4.2: Layout and cell view for synthesized circuit c432.



(a) Layout in GDSII format (viewed using Mentor Graphic Calibre Workbench)

```
DF;
DS 25 1 10;
9 INV8;
L L61;94 gnd! 60,180;94 vdd! 60,2400;94 A 140,1040;94 Q 980,1560;L L61;
94 NC1 1100,2610;94 INV8 20,2610;94 HIT-Kit:_Digital 120,20;
94 Owner:_austriamicrosystems 20,20;L L62;B 1120 2600 560,1300;L L35;P 1120,360
770,360 770,860 630,860 630,360 200,360 200,860 60,860 60,360 0,360 0,0 1120,0;
P 1120,2600 0,2600 0,2240 60,2240 60,1530 200,1530 200,2240 640,2240 640,1530
780,1530 780,2240 1120,2240;P 1070,1650 1060,1650 1060,2150 920,2150 920,1650
890,1650 890,1470 950,1470 950,1215 470,1215 470,1470 490,1470 490,2150
350,2150 350,1470 370,1470 370,770 360,770 360,450 500,450 500,770 470,770
470,1115 950,1115 950,770 920,770 920,450 1060,450 1060,770 1050,770 1050,1470
1070,1470;P 230,1130 210,1130 210,1240 70,1240 70,1130 50,1130 50,950 230,950;
L L34;B 80 80 990,700;B 80 80 990,1540;B 80 80 990,1720;B 80 80 990,1900;
B 80 80 990,2080;B 80 80 840,80;B 80 80 840,2520;B 80 80 710,1600;
B 80 80 710,1780;B 80 80 710,1960;B 80 80 710,2140;B 80 80 700,430;
B 80 80 700,610;B 80 80 700,790;B 80 80 990,520;B 80 80 560,80;B 80 80 560,2520;
B 80 80 430,520;B 80 80 430,700;B 80 80 420,1540;B 80 80 420,1720;
B 80 80 420,1900;B 80 80 420,2080;B 80 80 280,80;B 80 80 280,2520;
B 80 80 140,1170;B 80 80 130,430;B 80 80 130,610;B 80 80 130,790;
B 80 80 130,1600;B 80 80 130,1780;B 80 80 130,1960;B 80 80 130,2140;L L24;
P 1120,60 970,60 970,200 160,200 160,60 0,60 0,-60 1120,-60;P 1120,2530
990,2530 990,2380 140,2380 140,2530 0,2530 0,1270 1120,1270;L L23;P 1120,970
0,970 0,70 140,70 140,220 990,220 990,70 1120,70;P 1120,2660 0,2660 0,2540
160,2540 160,2400 970,2400 970,2540 1120,2540;L L20;P 890,2370 820,2370
820,1205 600,1205 600,2370 530,2370 530,1205 315,1205 315,2370 245,2370
245,1250 60,1250 60,1090 245,1090 245,265 315,265 315,1135 530,1135 530,265
600,265 600,1135 820,1135 820,265 890,265;L L10;P 60,345 1060,345 1060,880
60,880;B 1000 860 560,1860;P 210,2450 915,2450 915,2600 210,2600;
B 700 150 560,75;L L8;B 1480 1660 560,1950;L L5;B 1480 1660 560,1950;
DF;
DS 26 1 10;
9 SRAM_BITCELL;
L L61;94 Chan[1] 60,26780;94 PC 60,24960;94 C[0] 46045,20800;
94 C[1] 46045,12740;94 C[4] 60,21840;94 C[5] 60,28340;94 B[0] 46045,16380;
94 B[4] 60,16900;94 B[8] 46045,22100;94 A[0] 46045,19240;94 A[4] 60,18200;

B 660 100 19460,520;B 610 100 14115,16640;B 330 100 10895,19240;
B 330 100 13975,19760;B 610 100 20885,29640;B 100 360 21140,29770;
B 330 100 21255,29900;B 610 100 17805,26780;B 330 100 22705,29640;
B 330 100 33905,24440;B 610 100 33765,21580;B 330 100 33295,14300;
B 330 100 13135,11440;B 330 100 18455,29900;B 330 100 25735,29900;
B 1500 100 24920,40820;B 610 100 34275,29640;C25 R -1 0 T12040,28340;
C25 R -1 0 T19320,23140;C25 MX T29960,12740;C25 R -1 0 MX T19320,28340;
C24 T14560,28340;C24 MX T12880,23140;C24 R -1 0 MX T19320,23140;
C24 R -1 0 T13720,17940;C24 T12880,17940;C24 T32200,17940;C24 MX T11480,12740;
C24 T10920,17940;C24 R -1 0 MX T27440,12740;C24 MX T28840,12740;
C24 R -1 0 MX T13160,12740;C24 R -1 0 MX T26040,17940;C24 R -1 0 T18200,12740;
C24 R -1 0 T14000,28340;C24 MX T13720,23140;C24 T17360,17940;
C23 MX T13440,28340;C23 R -1 0 MX T24360,17940;C22 T10920,23140;
```

(b) Layout in CIF text format.

Figure 4.3: Layout in GDSII and CIF text format.

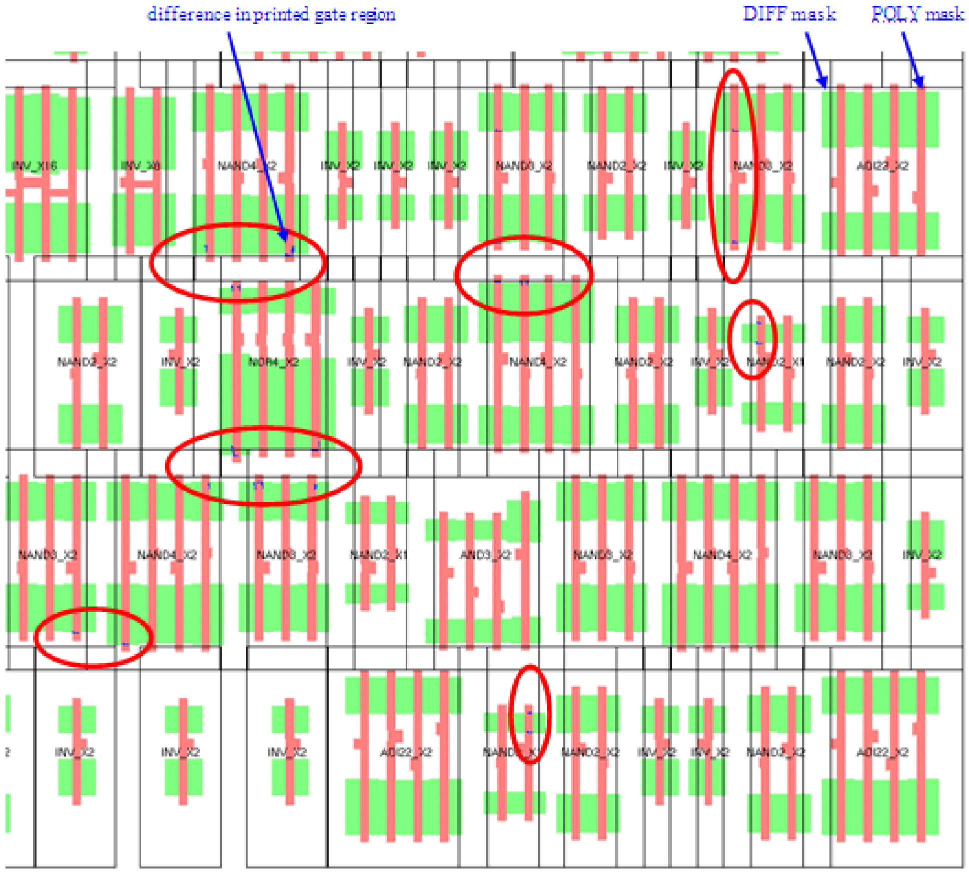


Figure 4.4: The differences in the post-lithography print image of gate regions (partial C432b circuit) are highlighted in blue region.

However, the environment perceived during library database construction (i.e. standard cell wise DPB-OPC) is different with its surrounding environment after placement. The proximity effects induced by different surrounding environment could affects the post-lithography print image and thereby the device performance. For instance, the difference in the post-lithography gate regions print image of partial C432b benchmark circuit is shown in Figure 4.4. As illustrated, not every boundary gate region get different print image than its counterpart of library database. In addition, the degree of distortion varies between transistors as some results in more or less slice transistor while other results in longer or shorter slice transistor. It is predictable that the former would results in relatively smaller performance disturbance than the later as *Ion* linearly dependent of *W* but exponentially varies with *L*.

The impacts of OPC induced performance error disturbance is then studied by simulate the print image for the entire layout and subsequently extract the performance error as post-placement *Ion* error, $Ion\_error_{post\text{-}placement}$. For each transistor, the *Ion* error changes (ΔIon_error) is measured by subtracting the standard cell wise *Ion* error (i.e. $Ion\_error_{cellwise}$ that stored in library database) from the post-placement *Ion* error, as shown in equation (4.1). A negative ΔIon_error value is desired as the performance errors get minimized after placement. Figure 4.5 shows the distribution of ΔIon_error of c432b circuit with an average of only 0.04%. Only 6% of the transistors (40 out of 662 transistors) exhibit increased *Ion* error with range of 0.05% to 7%. Figure 4.6 plots the % gate with poorer ΔIon_error and maximum (max) ΔIon_error for all ISCAS85' test cases. To rectify these negative OPE-caused performance shifts, localized DPB-OPC refinement step can then be performed.

$$\Delta Ion\_error = Ion\_error_{post-placement} - Ion\_error_{cellwise} \qquad (4.1)$$

The localized DPB-OPC refinement step is an adaptive correction attempted based on the diagnosed "device performance disturbance" displayed by ΔIon_error. The allowable maximum performance error shift is input by user as MAXDIFF. Transistors with ΔIon_error exceeding MAXDIFF will be tagged and subjected to localized refinement step. During this step, the respective transistor mask will be altered iteratively until the post-placement *Ion* error locally minimized.

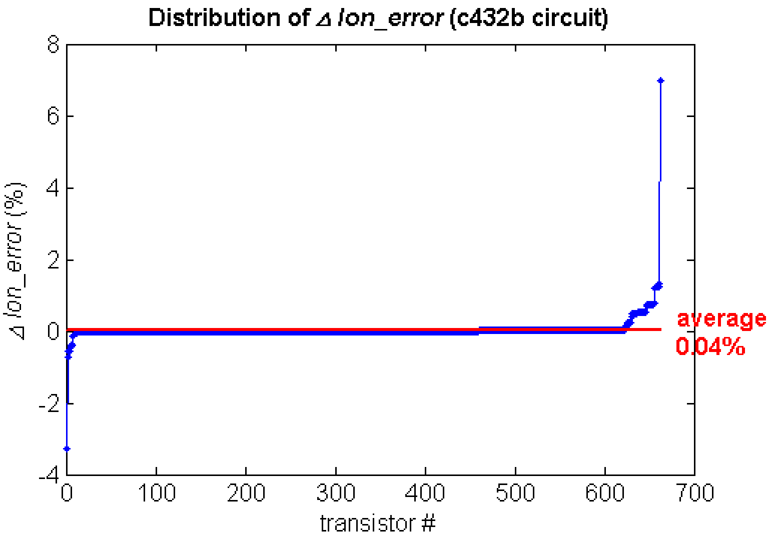


Figure 4.5: Distribution of Δ*Ion_error* of c432b circuit. The average Δ*Ion_error* is 0.04%.

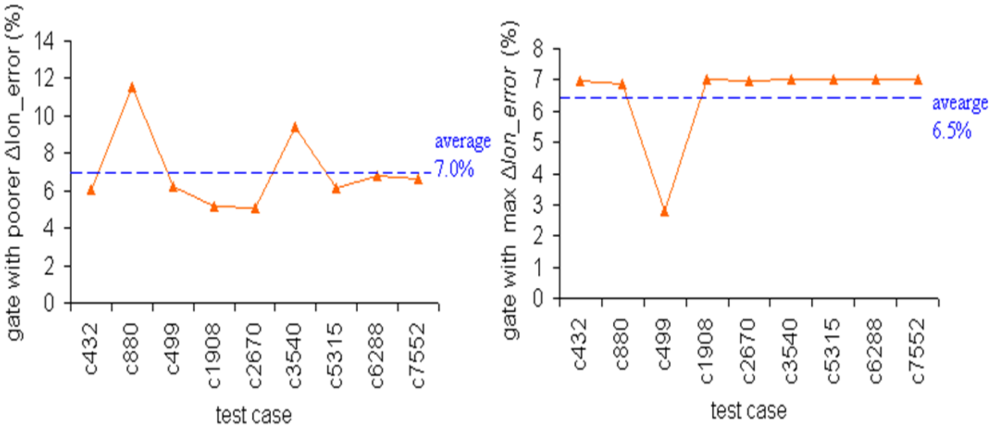


Figure 4.6: Plot of gate with poorer Δ*Ion_error* and max Δ*Ion_error* for all ISCAS85' test cases.

4.3 Results and Discussions

A simulation study is conducted on the IEEE International Symposium on Circuits and Systems 1985 (ISCAS' 85) benchmark circuits that synthesized based on 65nm standard cell library. The ISCAS' 85 circuits are synthesized from the Verilog list [66] using Synopsys Design Compiler version W2004.12-SP4. The synthesized netlists are then placed and routed using Cadence SOC_Encounter 7.1

[68] with instances from the 65nm library. The synthesized layouts with hierarchical information preserved are used as the test cases input to the proposed library-based DPB-OPC approach. The proposed library-based DPB-OPC flow is implemented in a Perl script and run in linux environment.

Two different OPC flows are examined and compared in terms of post-lithography device performance error and runtime:

- Full chip DPB-OPC flow, which correct the entire layout after placement. The hierarchical layout is flattened before subjected to this flow.
- Library-based DPB-OPC, which correct each standard cell offline with DPB-OPC framework and construct the entire mask by proper substitution and localized DPB-OPC refinement when necessary (which determined by MAXDIFF).

The chosen device performance error metrics are mean *Ion_error*, sigma of *Ion_error* distribution, maximum *Ion_error*, which defined as follows:

Let d_j denotes the *Ion_error* for transistor with index *j*,

mean *Ion* deviation,

$$\bar{d} = \frac{\sum d_j}{N} \qquad (4.2)$$

sigma of *Ion* deviation distribution,

$$\sigma_d = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(d_j - \bar{d}\right)^2} = \sqrt{\frac{1}{N}\left(\sum_{j=1}^{N} d_j{}^2\right) - \left(\bar{d}\right)^2} \qquad (4.3)$$

maximum *Ion* deviation,

$$d_{max} = \max_{\forall j} d_j \qquad (4.4)$$

Table 4.1 summarizes the comparison of post-OPC device performance error between full chip DPB-OPC and the proposed library-based DPB-OPC (with MAXDIFF=∞ to disable the localized DPB-OPC refinement). A smaller *Ion* error is desired as it indicates closer post-lithography device performance to the design value. It is interesting to find out that the library-based DPB-OPC approach generally capable of reducing the *Ion* error metrics, with an average of 13.6%, 18.9% and 14.2% reduction of mean, max and sigma of *Ion* error in the ISCAS'85 benchmark design.

Table 4.1: Comparison of post-OPC device performance error.

Circuit	Full chip DPB-OPC *Ion_error* (%)			Library-based DPB-OPC *Ion_error* (%)			Improvement in *Ion_error* |Library-based/Full chip| - 1 (%)		
	Mean	Max	Sigma	Mean	Max	Sigma	Mean	Max	Sigma
c432	1.34	14.81	2.24	1.58	12.16	2.46	17.4	-17.9	9.7
c880	0.87	11.15	1.50	0.84	7.33	1.51	-3.8	-34.3	1.0
c499	1.83	10.35	2.63	1.95	10.87	2.97	6.8	5.0	12.9
c1908	1.55	11.70	2.54	1.41	10.92	2.42	-9.1	-6.6	-5.1
c2670	0.85	12.66	1.71	0.48	7.28	0.98	-42.9	-42.5	-42.7
c3540	0.85	12.79	1.82	0.82	10.50	1.55	-3.6	-17.9	-14.6
c5315	0.81	12.79	1.71	0.57	10.92	1.20	-29.3	-14.6	-29.5
c6288	0.82	12.79	1.72	0.58	10.92	1.21	-29.0	-14.6	-29.7
c7552	0.81	14.97	1.72	0.58	11.01	1.21	-29.0	-26.5	-29.7
Average							-13.6	-18.9	-14.2

Table 4.2 shows the comparison of OPC run time. The library-based DPB-OPC approach (with localized refinement mode disabled) can reduce the OPC run time by up to 44× when compared to typical full chip DPB-OPC. The run time advantages would be substantially increased with more complex circuit.

Table 4.2: Comparison of run time.

Circuit	# Transistor	Run time (s)		
		Full chip DPB-OPC	Library-based DPB-OPC	Reduction (×)
c432	662	444.3	29.4	15.1
c880	1778	594.1	34.2	17.4
c499	1862	584.9	35.1	16.7
c1908	2032	685.5	35.8	19.1
c2670	4934	2025.7	68.1	29.7
c3540	6250	3843.0	87.4	44.0
c5315	10738	6639.56	198.4	33.5
c6288	10738	6645.5	194.4	34.2
c7552	10738	6273.5	196.5	31.9
Average				26.8

The above simulation is repeated with MAXDIFF = 2% and 0.01%. The reduction in *Ion* error metrics and run time improvements for all cases are plotted in Figure 4.7. Table 4.3 summarizes the average reduction achieved in the ISCAS' 85 test cases. The average improvement in *Ion* error metrics increases with smaller MAXDIFF but at the cost of decreasing run time reduction ratio. As shown in Figure 4.7 (d), the library-based DPB-OPC approach (with localized refinement mode enabled) can reduce the OPC run time by up to 5× when compared to typical full chip DPB-OPC.

Table 4.3: Comparison of run time for different MAXDIFF settings.

MAXDIFF (%)	Average reduction in *Ion_error* (%)			Average reduction in run time (×)
	Mean	Max	Sigma	
∞	13.6	18.9	14.2	26.8
2	16.1	18.9	16.4	3.2
0.01	21.5	20.3	19.7	3.2

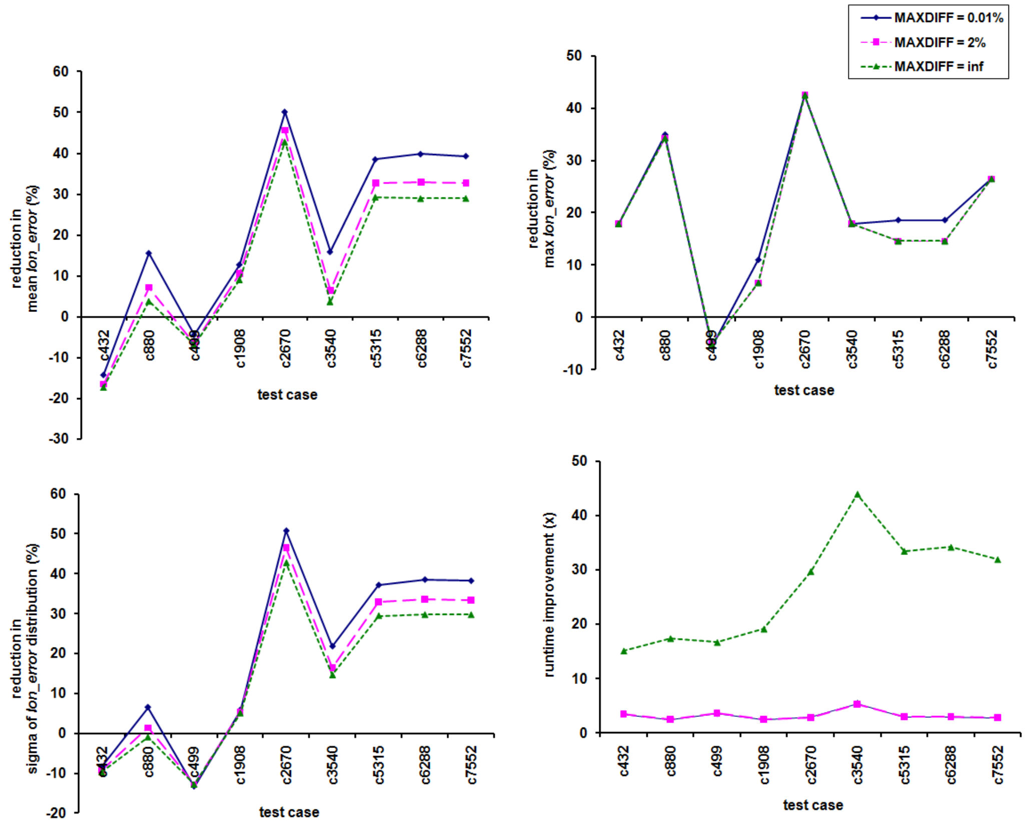


Figure 4.7: Reduction of *Ion* error metrics and run time improvements for MAXDIFF = ∞, 2% and 0.01%.

4.4 Chapter Summary

The proposed library-based DPB-OPC methodology has performance comparable to full chip DPB-OPC with significant run time reduction, up to 44× in the ISCAS’85 benchmark design. In addition, better performance matching was achieved in most test cases with library-based DPB-OPC approach. Based on the simulated performance disturbance map, the transistors with degraded *Ion* error can be fine-tuned by the adaptive correction step but at the expense of additional computational effort. This motivates the deployment of cell wise DPB-OPC as early as during standard cell library layout realization and characterization. On the other hand, the current standard cell wise DPB-OPC mask library creation was performed in the absence of representative environment or assist features, which results in average 7% transistors count with at most of 7% increase in *Ion* error. More suitable representative environment to reduce the discrepancy between the cell wise and the post-placement full chip approach can be explored further.

Chapter 5

Device Current and Capacitance Oriented OPC (IC-OPC)

5.1 Introduction

As mentioned in Chapter 3, DPB-OPC framework is proposed to synthesize simpler masks with printed patterns' *Ion* closely matching the designed value. However, it suffers larger gate capacitance deviation than the performance-optimized EPE-OPC due to the fact that DPB-OPC only targets at matching the desired *Ion* rather than the desired printed patterns. In order to achieve a balance between matching both drive current and gate area capacitance while reducing the mask complexity, an improved OPC approach namely IC-OPC is proposed in this chapter.

By simulation, the proposed IC-OPC outperforms the performance-optimized EPE-OPC approach in three aspects: an average of 32% reduction in mean path delay deviation, an average of 34% reduction in mask size and at least of 84% run time saving. Besides, IC-OPC also reduces the mean and standard deviation of normalized path delay deviation of DPB-OPC by average 7% and 2%.

Chapter 5 is organized as follows. Section 5.2 covers the proposed IC-OPC approach. Section 5.3 compares the simulation results between IC-OPC, conventional EPE-OPC and full chip DPB-OPC. Lastly, chapter summary is given in Section 5.4.

5.2 Overview of IC-OPC Flow

Figure 5.1 shows the proposed IC-OPC flow to synthesize polysilicon and diffusion masks. The inputs are the designed layout and reference library that provides necessary information for extracting post-lithography performance. First, the designed performance for each layout transistor - *Ion* and gate capacitance (*C*) are extracted as reference point. Then, the mask (which is the replica of designed layout) is subjected to lithography process simulator to generate the resulting on-wafer printed patterns. As a result of nonlinear pattern transfer during the lithography process, the printed wafer pattern differs from the original layout, causing the printed transistor performance to deviate from the designed value. Based on these printed patterns, the post-lithography drive current and gate capacitance of transistors are extracted using the models described in Section 2.2.4. The difference between the designed and post-OPC performance are calculated as follows.

$$\Delta I = \left| \frac{Ion_{post-lithography} - Ion_{design}}{Ion_{design}} \times 100\% \right| \tag{5.1}$$

$$\Delta C = \left| \frac{C_{post-lithography} - C_{design}}{C_{design}} \times 100\% \right| \tag{5.2}$$

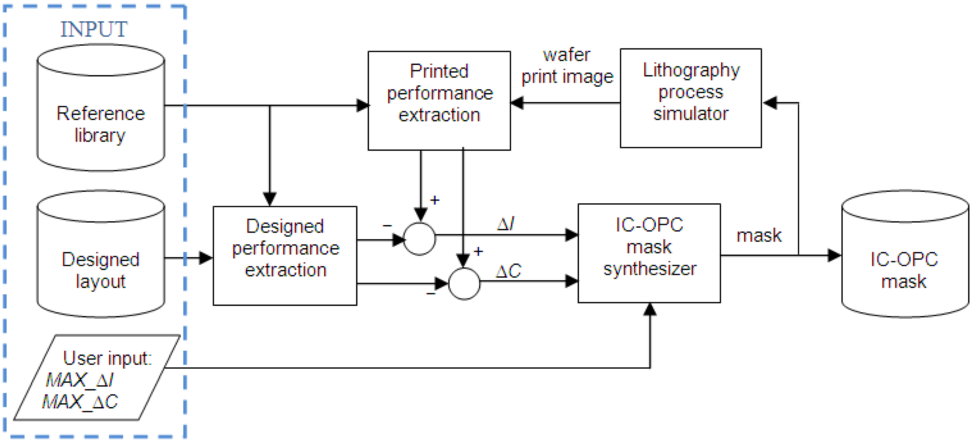


Figure 5.1: The proposed IC-OPC flow.

Guided by Δ*I* and Δ*C*, the mask will be corrected by the IC-OPC mask synthesizer as described in Section 5.2.1 until these performance deviations converges to the user-specified limit or reaches maximum iterations. The final mask is the IC-OPC mask with post-OPC performance resembling the designed value.

It has been shown in Section 3.3.3 that the gate capacitance can be modeled by the NRG_gate_area, hence the capacitance deviation ΔC can be indirectly measured through gate area deviation ΔA:

$$\Delta A = \left| \frac{A_{post-lithography} - A_{design}}{A_{design}} \times 100\% \right| \tag{5.3}$$

where

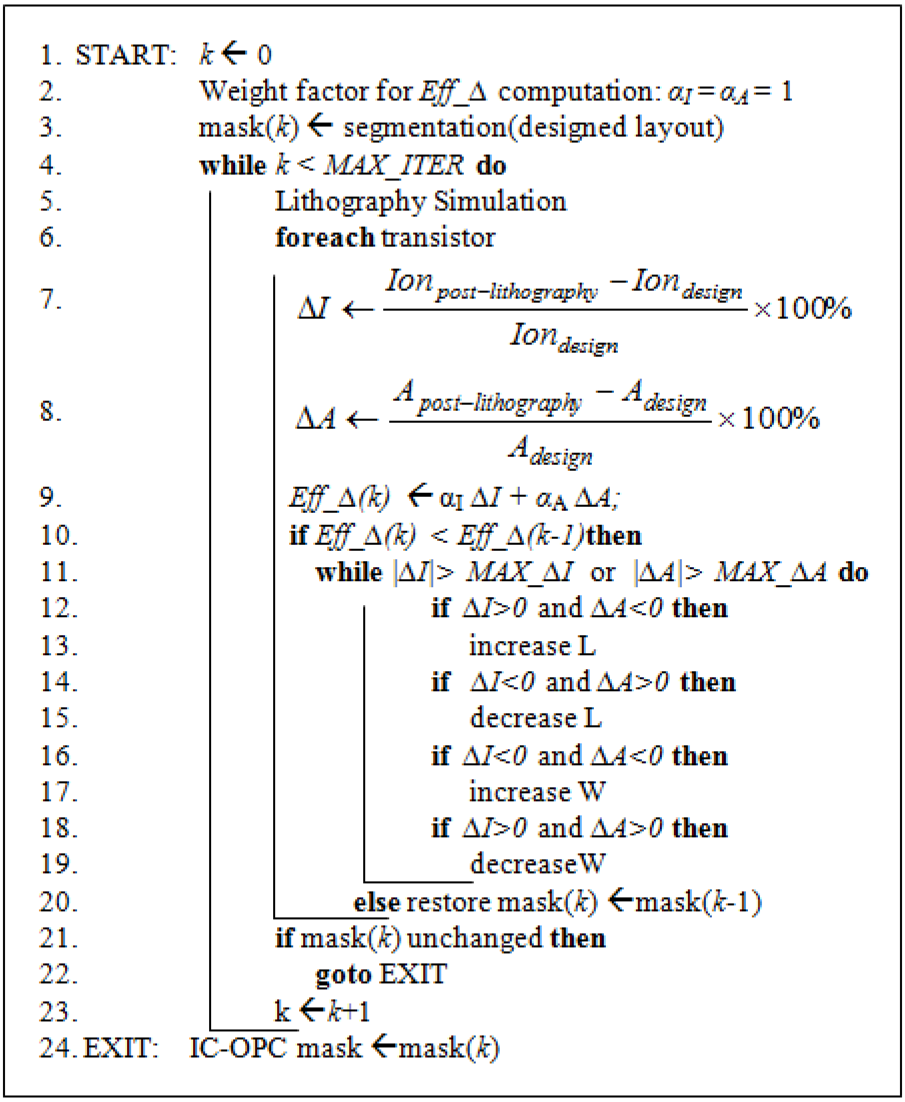
$$A_{post-lithoraphy} = \sum_{k=1}^{m} L_k W_k \tag{5.4}$$

5.2.1 IC-OPC Mask Synthesizer

The proposed mask synthesizer algorithm is outlined in Figure 5.2. Line 1 initializes the iteration counter *k* to zero. In line 3, segmentation process is performed on designed layout to dissect and tag the polysilicon and diffusion mask polygons for every transistor. The details for segmentation process have been provided in Section 3.2.2 *(1)*.

Next, the mask is subjected to lithography simulation in line 5. Then based on the on-wafer printed pattern, ΔI and ΔA for each transistor are computed. The effective deviation at iteration *k*, $Eff\_\Delta(k) = \alpha_I \Delta I + \alpha_A \Delta A$ will be evaluated and monitored in line 10–11 to ensure the effective deviation is minimized over consecutive iterations. This is achieved by reversing the mask correction for increasing *Eff*_Δ (line 20) while continuing the mask correction for decreasing *Eff*_Δ (line 12-19). Weighting factor α_I and α_A are introduced to enable user to set priority between the objective of minimizing current deviation and capacitance deviation. Subsequently if either Δ*I* or Δ*A* exceeds the user specified limit *MAX*_Δ*I* and *MAX*_Δ*A* respectively, mask correction routine (line 12-19) will be carried out on the corresponding transistor.

As long as the maximum iteration *MAX_ITER* is not reached in line 4, the updated mask will be again subjected to lithography simulation (line 5) and subsequent steps. The process repeats until the mask design converges (i.e. individual transistor's Δ*I,* Δ*A* fall below *MAX*_Δ*I, MAX*_Δ*A* or maximum iteration is reached.



```
START: $k \leftarrow 0$
       Weight factor for $Eff\_\Delta$ computation: $\alpha_I = \alpha_A = 1$
       mask($k$) $\leftarrow$ segmentation(designed layout)
       while $k < MAX\_ITER$ do
           Lithography Simulation
           foreach transistor
               $\Delta I \leftarrow \frac{Ion_{post-lithography} - Ion_{design}}{Ion_{design}} \times 100\%$
               $\Delta A \leftarrow \frac{A_{post-lithography} - A_{design}}{A_{design}} \times 100\%$
               $Eff\_\Delta(k) \leftarrow \alpha_I \Delta I + \alpha_A \Delta A;$
               if $Eff\_\Delta(k) < Eff\_\Delta(k\text{-}1)$ then
                 while $|\Delta I| > MAX\_\Delta I$ or $|\Delta A| > MAX\_\Delta A$ do
                       if $\Delta I > 0$ and $\Delta A < 0$ then
                           increase L
                       if $\Delta I < 0$ and $\Delta A > 0$ then
                           decrease L
                       if $\Delta I < 0$ and $\Delta A < 0$ then
                           increase W
                       if $\Delta I > 0$ and $\Delta A > 0$ then
                           decrease W
                   else restore mask($k$) $\leftarrow$ mask($k$-1)
           if mask($k$) unchanged then
               goto EXIT
           k $\leftarrow k$+1
EXIT:  IC-OPC mask $\leftarrow$ mask($k$)
```

Figure 5.2: The IC-OPC mask synthesizer algorithm.

The mask correction in L or W directions is achieved by stretching or compressing the segmented mask polygons in specific direction according to the decision matrix (Figure 5.3). The decision matrix is constructed based on the following three observations:

- *Ion* is linearly proportional to W but inversely proportional to L
- printed gate area is linearly proportionally to both W and L (Figure 5.4)

- printed gate region is assumed to be affected linearly by the mask polygon resizing operation

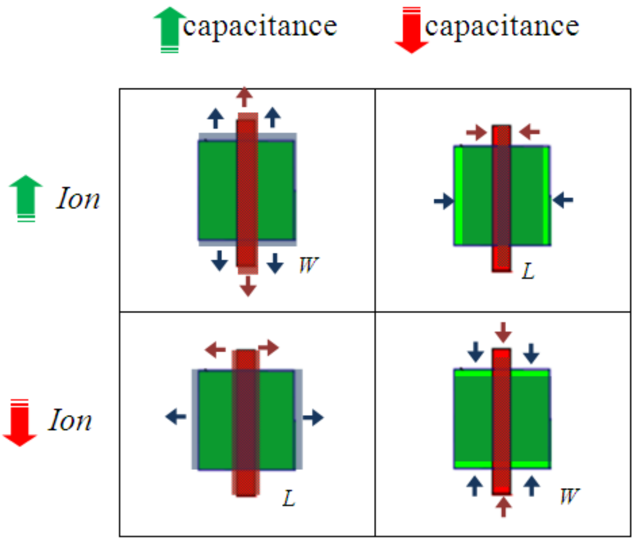


Figure 5.3: Decision matrix for mask correction.

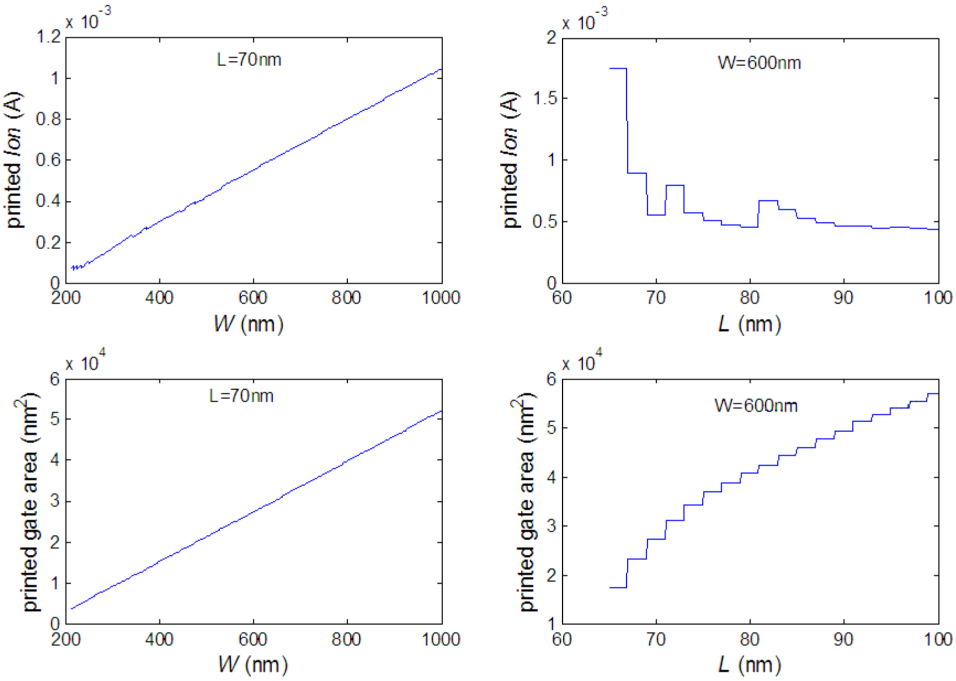


Figure 5.4: Effect of mask size changes on the printed *Ion* and gate area of an isolated transistor.

5.3 Results and Discussions

The proposed IC-OPC approach is implemented in Perl and tested on ISCAS85 benchmark circuits. These layouts are synthesized from Verilog files [66] and 65nm library using Synopsys Design Compiler version W2004.12-SP4 [67] and Cadence SOC_Encounter 7.1 [68].

The proposed IC-OPC approach is compared against the EPE-OPC approach and earlier work DPB-OPC [56] in following aspects: post-OPC performance deviation (drive current, gate area and path delay), mask size, and run time. For this purpose, the EPE-OPC mask is generated using Calibre OPCpro [69] with optimal OPCpro setting found from a rigorous search, which was conducted for fragmentation length of 10nm to 70nm in step of 2nm, iterations of 1 to 20 in step of 1, and step size of 1nm.

5.3.1 Post-OPC Performance Deviation

Figure 5.5 shows the mean and standard deviation (stdv) of drive current deviation and gate area deviation for all nine ISCAS85 benchmark circuits. The iso-trendline is where the mean or standard deviation of the two variables (drive current deviation and gate area deviation) equalizes. As indicated by the first order model of propagation delay [73], the propagation delay deviation can be approximated as following:

$$\Delta t_{pHL} = 0.69\frac{3C_L V_{DD}}{4Ion_n}(\Delta C_L / C_L - \Delta Ion_n / Ion_n) \tag{5.5}$$

$$\Delta t_{pLH} = 0.69\frac{3C_L V_{DD}}{4Ion_p}(\Delta C_L / C_L - \Delta Ion_p / Ion_p) \tag{5.6}$$

Furthermore, it has been shown that the capacitance deviation is closely related with the gate area deviation. Hence, it is desirable to have data closer to the iso-trendline (i.e. $\Delta C / C \approx \Delta Ion_p / Ion_p$) for smaller propagation delay deviation. It is clearly seen in Figure 5.5 that the proposed IC-OPC (with $\alpha_I = \alpha_A = 1$) outperforms the other approaches as its results are more closer to the iso-trendline. Due to limited computing resources, five out of nine benchmark circuits are randomly chosen for delay path extraction. These circuits are subjected to 22000 input transition patterns in SPICE simulation to extract the path delay.

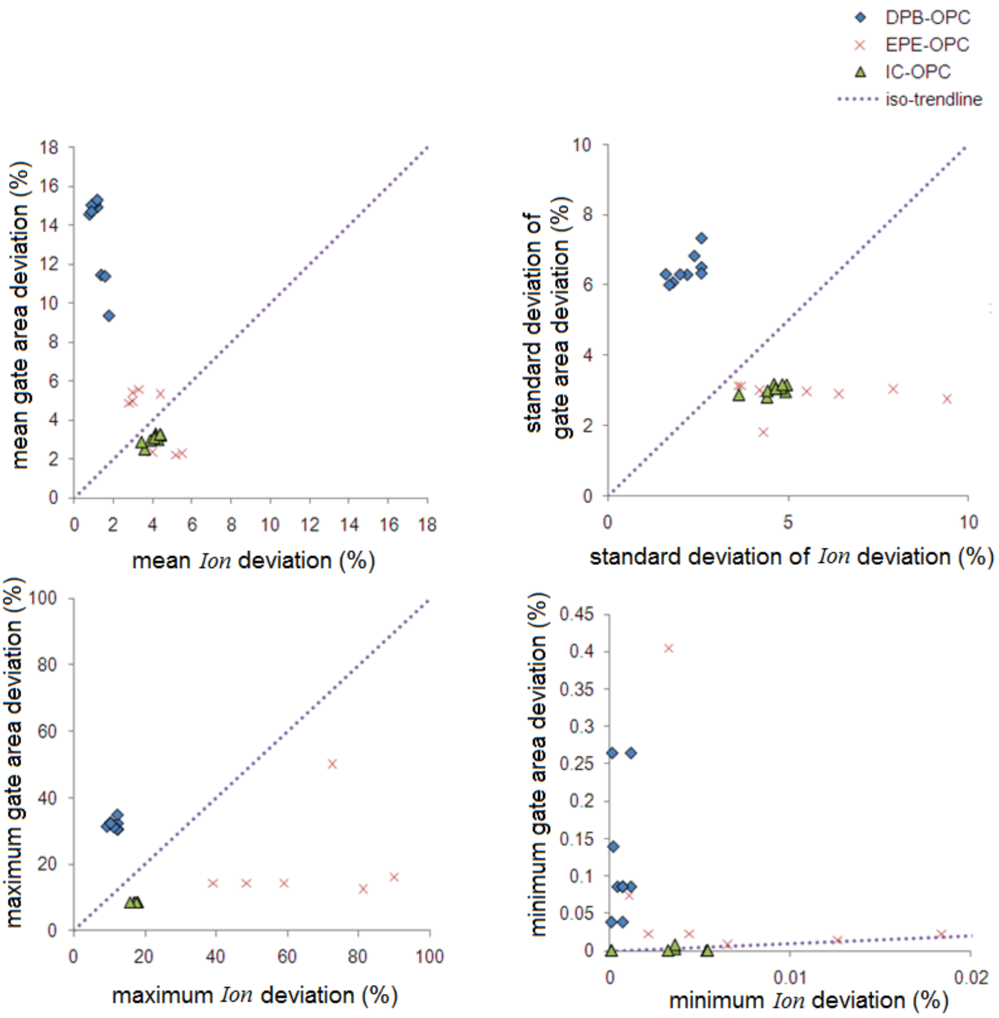


Figure 5.5: Mean and standard deviation of gate area deviation and *Ion* deviation for ISCAS85 test circuits.

Table 5.1: Normalized path delay deviation with respect to EPE-OPC.

Circuit	EPE-OPC		DPB-OPC		**IC-OPC**	
	mean	stdv	mean	stdv	mean	stdv
c432	1	1	0.50	0.43	0.55	0.47
c499	1	1	1.18	0.73	0.88	0.64
c880	1	1	0.75	0.71	0.83	0.75
c1908	1	1	1.05	1.09	0.72	0.87
c6288	1	1	0.61	1.05	0.59	1.05
average	1	1	0.75	0.82	0.68	0.80

Table 5.1 tabulates the normalized path delay deviation with respect to the EPE-OPC result. On average, the proposed IC-OPC reduced the mean by 32% and standard deviation by 20% when compared to EPE-OPC. It is worth pointing out that the IC-OPC also improved the DPB-OPC by reducing the mean and standard deviation to less than normalized value 1.0. With path delay deviation normalized with respect to EPE-OPC, IC-OPC reduces the mean and standard deviation of DPB-OPC by additional absolute value of 7% and 2% respectively. Such improvement can be illustrated in the histograms of path delay deviation for post-OPC c1908 circuits (Figure 5.6). The path delay deviation is improved from 2.43% (DPB-OPC) to 1.67% (IC-OPC) due to better co-matching of *Ion* and gate capacitance (i.e. closer to iso-trendline in Figure 5.5). Furthermore, it is interesting to observe that the IC-OPC could suppress the gate area deviation trendline to be lower than that of EPE-OPC across the transistor width range in circuit c1908 (Figure 5.7). The histograms of path delay deviation for the remaining 4 circuits are provided in Figure 5.8.

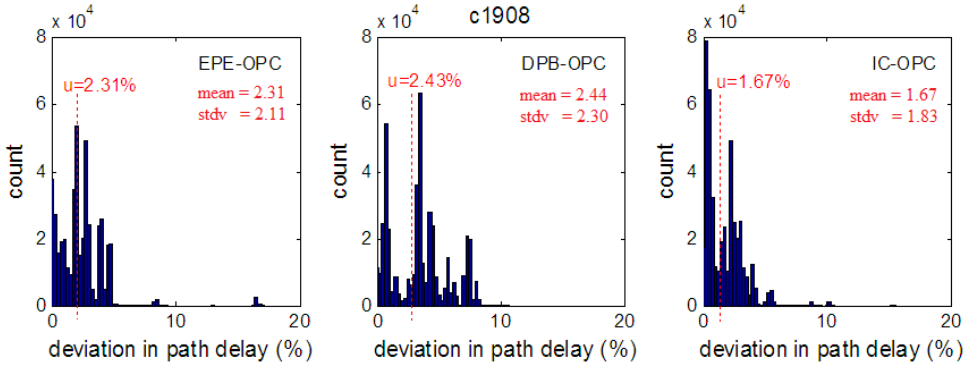


Figure 5.6: Histogram of path delay deviation for post-OPC c1908 circuit.

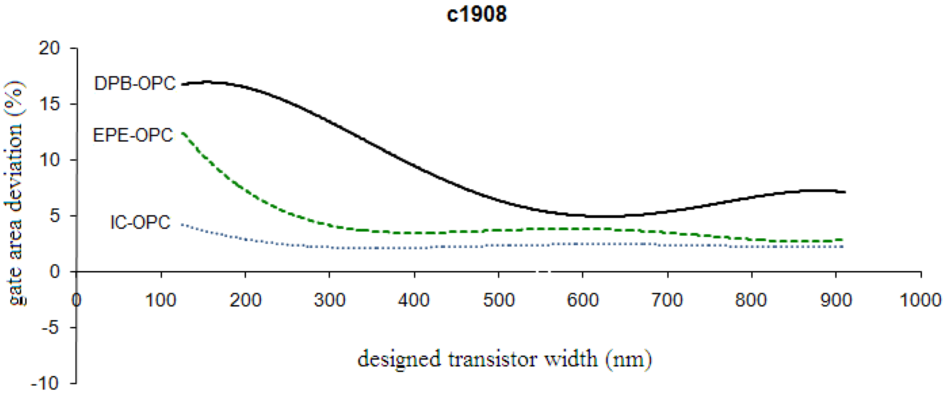


Figure 5.7: Trendline relationship between the mean gate area deviation and the designed transistor width.

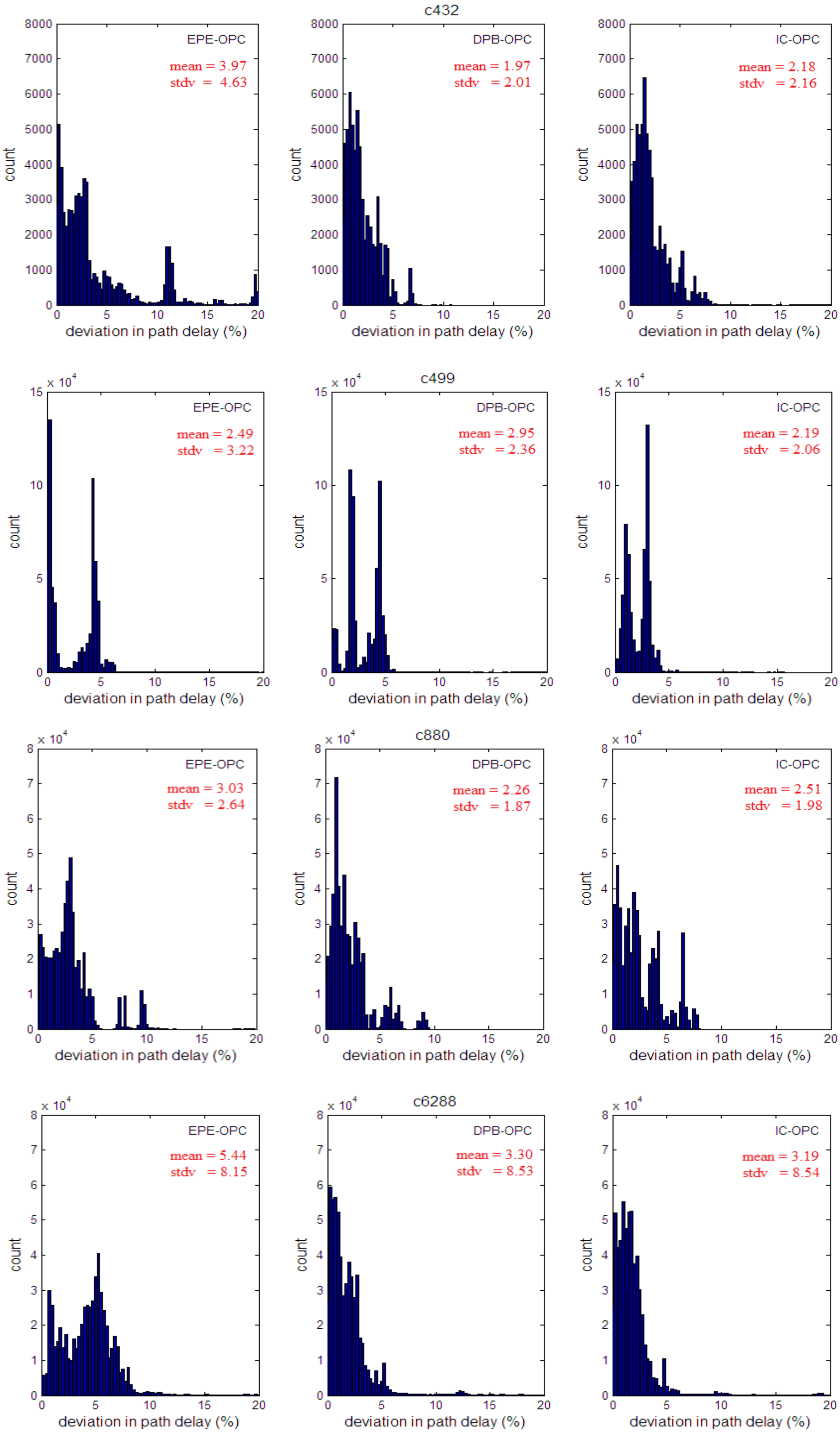


Figure 5.8: The histogram of path delay deviation for c432, c499, c880 and c6288.

5.3.2 Mask Size

Table 5.2 summarizes the normalized mask size for both diffusion and polysilicon layers for all three OPC approaches. With IC-OPC, mask complexity are reduced by 17% (diffusion mask) and 57% (polysilicon mask) on average when compared to EPE-OPC. Then, the mask size for both DPB- and IC- are comparable as expected due to the similar mask synthesizing concept and DRC regulator employed in the flow.

Table 5.2: Normalized mask size with respect to EPE-OPC.

Circuit	EPE-OPC		DPB-OPC		IC-OPC	
	Diffusion	Polysilicon	Diffusion	Polysilicon	Diffusion	Polysilicon
c432	1	1	0.93	0.48	0.93	0.48
c499	1	1	0.80	0.39	0.77	0.39
c880	1	1	0.88	0.50	0.85	0.46
c1908	1	1	0.91	0.46	0.91	0.41
c2670	1	1	0.88	0.52	0.88	0.46
c3540	1	1	0.80	0.50	0.80	0.46
c5315	1	1	0.85	0.52	0.84	0.46
c6288	1	1	0.76	0.36	0.75	0.32
c7552	1	1	0.80	0.52	0.85	0.46
average	1	1	0.85	0.47	0.84	0.43

5.3.3 Run Time

The run time require for both IC-OPC and DPB-OPC flows are normalized against the EPE-OPC search time in Table 5.3. When compared to EPE-OPC, at least of 84% run time saving is achieved in IC-OPC; this is an additional 25% reduction in run time when compared to DPB-OPC. EPE-OPC is slowest due to the search time taken to search the optimal OPCpro setting for best *Ion* performance. IC-OPC is faster than DPB-OPC due to the difference in the decision matrix for mask correction. For DPB-OPC, coarse tuning by modifying L is followed by fine tuning with W and fine tuning step needs most iterations. But for IC-OPC, there is no fine tuning involved.

Table 5.3: Normalized run time with respect to EPE-OPC search time.

Circuit	EPE-OPC	DPB-OPC	IC-OPC
c432	1	0.083	0.016
c499	1	0.073	0.027
c880	1	0.415	0.159
c1908	1	0.055	0.010
c2670	1	0.051	0.014
c3540	1	0.065	0.014
c5315	1	0.092	0.032
c6288	1	0.049	0.024
c7552	1	0.094	0.039
average	1	0.108	0.037

5.4 Chapter Summary

This chapter proposes an IC-OPC to co-optimize the post-OPC circuit performance, i.e. both drive current and gate are capacitance, and mask size. IC-OPC synthesizes simpler masks such that the estimated post-lithography drive current and gate area capacitance resemble the designed layout. As a result, IC-OPC (with $\alpha_I = \alpha_A = 1$) achieves 32% reduction in mean path delay deviation, 37% reduction in mask size and at least of 84% run time saving when compared to the performance-optimized EPE-OPC. On average, IC-OPC achieves additional absolute 7% and 2% reduction in mean and standard deviation of normalized path delay deviation than DPB-OPC.

Chapter 6

Conclusion

6.1 Summary

This thesis examines the application of design-process integration concept into the OPC mask design problem to meet the challenge of cost control on resolution enhanced optical mask.

In Chapter 2, a PB-OPC framework is presented to generate simpler OPC mask that achieves closer circuit performance matching. The proposed framework exploits the design intent extractable from the design layout to guide upon the customized OPC mask generator. Simulation results shows that the proposed approach can achieve great saving in mask data volume and closer circuit performance matching to the design intent.

In Chapter 3, a generalized DPB-OPC framework with a few improvements is developed. Firstly, a weighting function $\gamma_k(w)$ is augmented to the gate-slicing model [53] used in the framework. This is to account for the non-linear current density along the channel width due to threshold voltage variation and edge effect. Secondly, the initial mask adjustment step of the mask design algorithm is pre-characterized to speed up the computation and this has results in an average of 3.07% further reduction in mean *Ion* deviation. Thirdly, a modular block called DRC compliance regulator is introduced to ensure that the OPC mask compliant to DRC rules and the post-OPC printed patterns are free from bridging, pinching, open or short issues. By simulation, DPB-OPC framework outperforms the performance-optimized EPE-OPC approach in two aspects: an average of 34% reduction in mask size and up to 13.5% reduction in device performance deviation.

Although the results are promising, the developed framework can be further improved in two perspectives: reduced OPC run time and better performance matching efficiency. Hence, a library-based cell wise DPB-OPC framework is developed in Chapter 4 to handle the synthesized digital circuit. By making use the hierarchical information of the synthesized circuit and the pre-characterized DPB-OPC library, the OPC run time efficiency can be greatly improved. Simulation demonstrates that the library-based performance-based OPC approach has performance comparable to full chip performance-based OPC and with significant run time reduction, up to 44× in the ISCAS'85 benchmark design. In addition, the transistors with degraded *Ion* error can be further fine-tuned by the adaptive correction step but at the expense of additional computational effort.

To achieve satisfactory co-matching on both *Ion* and gate capacitance, a hybrid IC-OPC correction algorithm is developed in Chapter 5. There are two main differences introduced: Firstly, the performance deviation error is the weighted sum of drive current and gate capacitance error. Secondly, decision matrix is constructed based on the relationship between *Ion*, gate area with respect to channel width and length under the assumption that the printed pattern circuit will be affected linearly with the mask resize amplitude. By simulation, the proposed IC-OPC outperforms the performance-optimized EPE-OPC approach in three aspects: an average of 32% reduction in mean path delay deviation, an average of 34% reduction in mask size and at least of 84% run time saving.

In sum, pros and cons for the developed frameworks are summarized in the Table 6.1.

Table 6.1: Comparison of various OPC frameworks.

	EPE-OPC	PB-OPC	DPB-OPC
PB-OPC	**Test cases**: NMOS, PMOS, inverter chain, SRAM **Post-OPC simulation**: SPICE + *LeqIon* **PB-OPC VS EPE-OPC:** ~ 33% mask size saving ~11% performance matching improvement		
DPB-OPC	**Test cases**: 65nm library standard cells, ISCAS85 circuits **Post-OPC simulation**: *Ion* deviation, SPICE + *LeqIon* for *tr, tf, tp, td* (+NRG capacitance). **DPB-OPC VS EPE-OPC:** ~ reduce absolute 1.7-3.7 % in mean *Ion* deviation ~ 33.3% mask size reduction ~ c1908: slight worst path delay due to larger capacitance deviation.	**Test cases**: 65nm library standard cells **Post-OPC circuit simulation**: *Ion* deviation **DPB-OPC VS PB-OPC:** ~ reduce 3.07% in mean *Ion* deviation ~ new features: Safety margin and weighted gate slicing model, characterizsation for init adjust, DRC compliance regulator	
Library-based DPB-OPC			**Test cases**: 65nm library standard cells, ISCAS85 circuits **Post-OPC simulation**: *Ion* deviation **Library-based DPB-OPC VS DPB-OPC:** ~ reduce average 13.6% in mean *Ion* deviation ~ OPC run time reduced 26.8 X ~ adjustable *MAXDIFF* for performance -runtime trade off
IC-OPC	**Test cases**: ISCAS85 circuits **Post-OPC simulation**: *Ion* deviation, gate area deviation, SPICE + *LeqIon* for td (+NRG capacitance). **IC-OPC VS EPE-OPC:** ~ (*Ion*,A) deviations closer to iso-trendline ~ improve path delay deviation by 33% ~46% mask size reduction		**Test cases**: ISCAS85 circuits **Post-OPC simulation**: *Ion* deviation, gate area deviation, SPICE + *LeqIon* for td (+NRG capacitance). **IC-OPC VS DPB-OPC:** ~ (*Ion*,A) deviations closer to iso-trendline ~ improve path delay deviation by absolute 7% ~comparable mask size due to similar mask synthesizing concept.

6.2 Future Work

As far as the thesis is concerned, the key idea behind the proposed performance-based OPC works is to leverage design intent information into the customized OPC mask design algorithm. The benefit of such methodologies in achieving mask size reduction as well as better post-lithography performance matching with designed value have been demonstrated. One possible future work is to extend similar concept to the other layers by considering the relevant performance requirement, such as RC delay for the backend interconnect layers.

Besides, it would be of interest to study the possibility of modeling and optimization of the performance-based OPC framework. The objective of such study is to synthesize the performance-based OPC mask, which is globally-optimized or otherwise sub-optimal, without the need of iterative lithography simulation. One possible way of formulating the performance-based OPC mask optimization problem is to combine the lithography modeling of partially coherent imaging system and the non-rectangular transistors modeling (Figure 6.1). As an OPC mask consists of only chrome and quartz features, the mask transmission values can be restricted to be either 0 or 1. Therefore the optimization problem will therefore be subjected to the constraints given by the allowable transmission values of binary 0 or 1. In addition, the allowable mask changes – mere resize of associated mask polygons - would also be incorporated as constraints to the optimization problem. This is to align with the algorithm of the proposed DPB-OPC framework which serves to control the mask complexity without compromising the performance degradation.

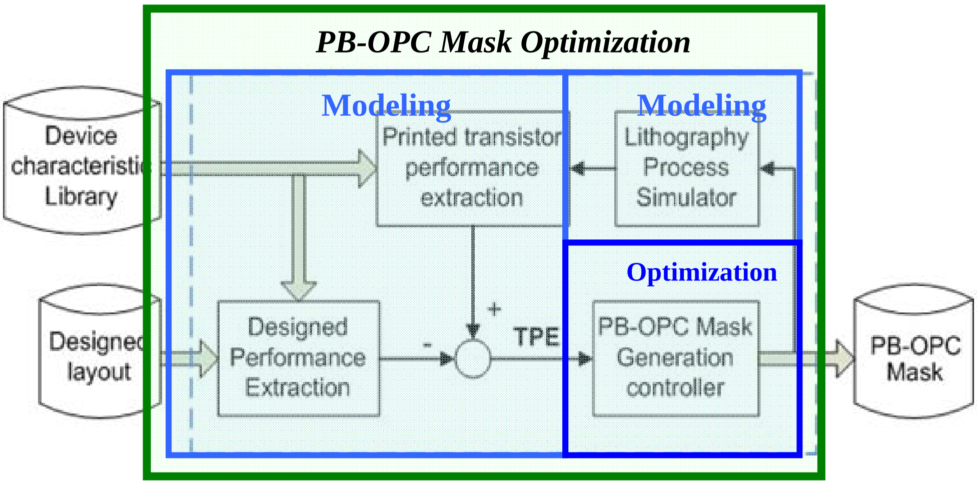

PB-OPC Mask Optimization
Modeling
Modeling
Device characteristic Library
Printed transistor performance extraction
Lithography Process Simulator
Optimization
Designed layout
Designed Performance Extraction
+
-
TPE
PB-OPC Mask Generation controller
PB-OPC Mask

Figure 6.1: Formulation of PB-OPC mask optimization problem.

Author's Publications

Journal Publications

1. S.-H. Teh, C.-H. Heng, and A. Tay, "Performance-based optical proximity correction methodology," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol 29, no. 1, pp. 51-64, 2010.

2. S. H. Teh, C. H. Heng, and A. Tay, "Adaptive library-based device performance-driven optical proximity correction," *Electronics Letters*, vol 46, no. 7, pp. 513-515, 2010.

Conference Publications

1. S.-H. Teh, C.-H. Heng, and A. Tay, "Design-process integration for performance-based OPC framework," in *Proc. ACM/IEEE Design Automation Conf*erence, Anaheim, CA, USA, 2008, pp. 522-527.

2. S.-H. Teh, C.-H. Heng, and A. Tay, "Device performance-based OPC for optimal circuit performance and mask cost reduction," in *Proc. of SPIE* vol. 6925, Santa Clara, CA, USA, 2008, pp. 692511.

3. S.-H. Teh, C.-H. Heng, and A. Tay, "Library-based performance-based OPC," in *Proc. of SPIE* vol. 7641, San Jose, CA, USA, 2010, pp. 76410X.

4. Y. Qu, S.H. Teh, C.-H. Heng, A. Tay and T.H. Lee, “Timing Performance Oriented Optical Proximity Correction for Mask Cost Reduction,” in *Proc. IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, San Francisco, CA, 2010, pp. 99-103.

Bibliography

[1] C. A. Mack, *Fundamental Principles of optical lithography : the science of microfabrication*. Chichester, West Sussex, England ; Hoboken, NJ, USA: Wiley, 2007.

[2] J. D. Plummer, *Silicon VLSI technology : fundamentals, practice and modeling*. Upper Saddle River, NJ: Prentice Hall, 2000.

[3] K. Suzuki and B. W. Smith, *Microlithography science and technology*, 2nd ed. Boca Raton: Taylor & Francis, 2007.

[4] International Technology Roadmap for Semiconductors 2009 [Online]. Available: http://www.itrs.net/Links/2009ITRS/Home2009.htm

[5] M. Quirk, *Semiconductor manufacturing technology*. Upper Saddle River, NJ: Prentice Hall, 2001.

[6] H. J. Levinson, *Principles of lithography*, 2nd ed. Bellingham, WA: SPIE Press, 2005.

[7] T. Ito and S. Okazaki, "Pushing the limits of lithography," *Nature,* vol. 406*,* no. 6799, pp. 1027-1031, 2000.

[8] International Technology Roadmap for Semiconductors 2009 Edition - Lithography [Online]. Available: http://www.itrs.net/Links/2009ITRS/2009Chapters_2009Tables/2009_Litho.pdf

[9] A. K.-K. Wong, *Resolution enhancement techniques in optical lithography*. Bellingham, WA: SPIE Press, 2001.

[10] C. Spence, "Full-chip lithography simulation and design analysis: how OPC is changing IC design," in *Proc. of SPIE* vol. 5751*,* San Jose, CA, USA, 2005, pp. 1-14.

[11] P. Gupta, A. B. Kahng, D. Sylvester, and J. Yang, "Performance-driven optical proximity correction for mask cost reduction," *Journal of Micro/ Nanolithography, MEMS, and MOEMS,* vol. 6*,* no. 3, p. 031005, 2007.

[12] Various Techniques for Achieving Desired CD Control and Overlay with Optical Projection Lithography for MPU and DRAM [Online]. Available: http://www.itrs.net/Links/2009ITRS/2009Chapters_2009Tables/2009Tables_LITH1.xls

[13] F. M. Schellenberg, H. Zhang, and J. Morrow, "SEMATECH J111 project: OPC validation," in *Proc. of SPIE* vol. 3334*,* Santa Clara, CA, USA, 1998, pp. 892-911.

[14] N. B. Cobb, "Fast optical and process proximity correction algorithms for integrated circuit manufacturing," Ph.D., University of California, Berkeley, California, United States, 1998.

[15] O. W. Otto, J. G. Garofalo, K. K. Low, C.-M. Yuan, R. C. Henderson, C. Pierrat, R. L. Kostelak, S. Vaidya, and P. K. Vasudev, "Automated optical proximity correction-a rules-based approach," in *Proc. of SPIE* vol. 2197*,* San Jose, CA, USA, 1994, pp. 278-293.

[16] D. M. Newmark, "Optical proximity correction for resolution enhancement technology," Ph.D., University of California, Berkeley, California, United States, 1994.

[17] Y. Liu and A. Zakhor, "Binary and phase shifting mask design for optical lithography," *IEEE Transactions on Semiconductor Manufacturing,* vol. 5*,* no. 2, pp. 138-152, 1992.

[18] Y. Liu, A. Pfau, and A. Zakhor, "Systematic design of phase-shifting masks with extended depth of focus and/or shifted focus plane," *IEEE Transactions on Semiconductor Manufacturing,* vol. 6*,* no. 1, pp. 1-21, 1993.

[19] Y. Liu, A. Zakhor, and M. A. Zuniga, "Computer-aided phase shift mask design with reduced complexity," *IEEE Transactions on Semiconductor Manufacturing,* vol. 9*,* no. 2, pp. 170-181, 1996.

[20] J. P. Stirniman and M. L. Rieger, "Spatial-filter models to describe IC lithographic behavior," in *Proc. of SPIE* vol. 3051*,* Santa Clara, CA, USA, 1997, pp. 469-478.

[21] J. P. Stirniman, M. L. Rieger, and R. Gleason, "Quantifying proximity and related effects in advanced wafer processes," in *Proc. of SPIE* vol. 2440*,* Santa Clara, CA, USA, 1995, pp. 252-260.

[22] J. P. Stirniman and M. L. Rieger, "Fast proximity correction with zone sampling," in *Proc. of SPIE* vol. 2197*,* San Jose, CA, USA, 1994, pp. 294-301.

[23] N. B. Cobb and A. Zakhor, "Fast sparse aerial-image calculation for OPC," in *Proc. of SPIE* vol. 2621*,* Santa Clara, CA, USA, 1995, pp. 534-545.

[24] N. B. Cobb and A. Zakhor, "Fast, low-complexity mask design," in *Proc. of SPIE* vol. 2440*,* Santa Clara, CA, USA, 1995, pp. 313-327.

[25] N. B. Cobb and A. Zakhor, "Large area phase-shift mask design," in *Proc. of SPIE* vol. 2197*,* San Jose, CA, USA, 1994, pp. 348-60.

[26] D. Z. Pan, P. Yu, M. Cho, A. Ramalingam, K. Kim, A. Rajaram, and S. X. Shi, "Design for manufacturing meets advanced process control: A survey," *Journal of Process Control,* vol. 18*,* no. 10, pp. 975-984, 2008.

[27] B. P. Wong, A. Mittal, G. W. Starr, F. Zach, V. Moroz, and A. Kahng, *Nano-CMOS Design for Manufacturability: Robust Circuit and Physical Design for Sub-65nm Technology Nodes*: Wiley-Interscience, 2008.

[28] A. M. B. Wong, Y. Cao, and G. W. Starr, *Nano-CMOS circuit and physical design*, 1st ed. Hoboken, N.J.: Wiley-Interscience, November 2004.

[29] International Technology Roadmap for Semiconductors 2008 [Online]. Available: http://www.itrs.net/Links/2008ITRS/Home2008.htm

[30] P. Gupta, A. B. Kahng, D. Sylvester, and J. Yang, "A cost-driven lithographic correction methodology based on off-the-shelf sizing tools," in *40th Design Automation Conference,* Anaheim, CA, United States, 2003, pp. 16-21.

[31] A. B. Kahng, S. Muddu, and C.-H. Park, "Auxiliary pattern-based optical proximity correction for better printability, timing, and leakage control," *Journal of Microlithography, Microfabrication, and Microsystems,* vol. 7*,* no. 1, pp. 013002-1, 2008.

[32] P. Gupta, F.-L. Heng, and M. Lavin, "Merits of cellwise model-based OPC," in *Proc. of SPIE* vol. 5379*,* Santa Clara, CA, United States, 2004, pp. 182-189.

[33] W. J. Trybula, "Cost of ownership - projecting the future," *Microelectronic Engineering,* vol. 83*,* no. 4-9 SPEC ISS, pp. 614-618, 2006.

[34] W. J. Trybula, "A common base for mask cost of ownership," in *Proc. of SPIE* vol. 5256*,* Monterey, CA, USA, 2003, pp. 318-23.

[35] Lithography CoO Analysis [Online]. Available: http://www.sematech.org

[36] A. Gu and A. Zakhor, "Optical proximity correction with linear regression," *IEEE Transactions on Semiconductor Manufacturing,* vol. 21*,* no. 2, pp. 263-71, 2008.

[37] S. Banerjee, P. Elakkumanan, L. W. Liebmann, J. A. Culp, and M. Orshansky, "Electrically driven optical proximity correction," in *Proc. of SPIE* vol. 6925*,* San Jose, CA, USA, 2008, pp. 69251-1.

[38] S. Banerjee, P. Elakkumanan, L. W. Liebmann, and M. Orshansky, "Electrically driven optical proximity correction based on linear programming," in *IEEE/ACM International Conference on Computer-Aided Design,* Piscataway, NJ, USA, 2008, pp. 473-479.

[39] C. Li, L. S. Milor, C. H. Ouyang, W. Maly, and Y.-K. Peng, "Analysis of the impact of proximity correction algorithms on circuit performance," *IEEE Transactions on Semiconductor Manufacturing,* vol. 12*,* no. 3, pp. 313-22, 1999.

[40] 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,* vol. 21*,* no. 5, pp. 544-53, 2002.

[41] M. Orshansky, L. Milor, and C. Hu, "Characterization of spatial intrafield gate CD variability, its impact on circuit performance, and spatial mask-

level correction," *IEEE Transactions on Semiconductor Manufacturing,* vol. 17*,* no. 1, pp. 2-11, 2004.

[42] M. Choi and L. Milor, "Impact on circuit performance of deterministic within-die variation in nanoscale semiconductor manufacturing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 25*,* no. 7, pp. 1350-1367, 2006.

[43] M.-F. You, P. C. W. Ng, Y.-S. Su, K.-Y. Tsai, and Y.-C. Lu, "Impacts of optical proximity correction settings on electrical performance," in *Proc. of SPIE* vol. 6521*,* San Jose, CA, United States, 2007, p. 65210.

[44] Y. Liu and J. Hu, "A New Algorithm for Simultaneous Gate Sizing and Threshold Voltage Assignment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 29*,* no. 2, pp. 223-234, 2010.

[45] A. Sanyal, A. Rastogi, W. Chen, and S. Kundu, "An Efficient Technique for Leakage Current Estimation in Nanoscaled CMOS Circuits Incorporating Self-Loading Effects," *IEEE Transactions on Computers,* vol. 59*,* no. 7, pp. 922-932, 2010.

[46] A. A. Bayrakci, A. Demir, and S. Tasiran, "Fast Monte Carlo Estimation of Timing Yield With Importance Sampling and Transistor-Level Circuit Simulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 29*,* no. 9, pp. 1328-1341, 2010.

[47] Q. Ding, Y. Wang, H. Wang, R. Luo, and H. Yang, "Output remapping technique for critical paths soft-error rate reduction," *Computers & Digital Techniques, IET,* vol. 4*,* no. 4, pp. 325-333, 2010.

[48] B. Bosio, P. Girard, S. Pravossoudovitch, and A. Virazel, "A Comprehensive Framework for Logic Diagnosis of Arbitrary Defects," *IEEE Transactions on Computers,* vol. 59*,* no. 3, pp. 289-300, 2010.

[49] Z. Feng and P. Li, "Performance-Oriented Parameter Dimension Reduction of VLSI Circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 17*,* no. 1, pp. 137-150, 2009.

[50] Z. Jiang and S. K. Gupta, "Threshold Testing: Improving Yield for Nanoscale VLSI," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 28*,* no. 12, pp. 1883-1895, 2009.

[51] T.-H. Wu and A. Davoodi, "PaRS: Parallel and Near-Optimal Grid-Based Cell Sizing for Library-Based Design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 28*,* no. 11, pp. 1666-1678, 2009.

[52] H. Jeon, Y.-B. Kim, and M. Choi, "Standby Leakage Power Reduction Technique for Nanoscale CMOS VLSI Systems," *IEEE Transactions on Instrumentation and Measurement,* vol. 59*,* no. 5, pp. 1127-1133, 2010.

[53] W. J. Poppe, L. Capodieci, J. Wu, and A. Neureuther, "From poly line to transistor: Building BSIM models for non-rectangular transistors," in *Proc. of SPIE* vol. 6156*,* San Jose, CA, United States, 2006, p. 61560.

[54] P. Gupta, A. Kahng, Y. Kim, S. Shah, and D. Sylvester, "Modeling of non-uniform device geometries for post-lithography circuit analysis," in *Proc. of SPIE* vol. 6156*,* San Jose, CA, United States, 2006, p. 61560.

[55] R. Singhal, A. Balijepalli, A. Subramaniam, F. Liu, S. Nassif, and Y. Cao, "Modeling and analysis of non-rectangular gate for post-lithography

circuit simulation," in *44th Design Automation Conference,* San Diego, CA, United States, 2007, pp. 823-828.

[56] S.-H. Teh, C.-H. Heng, and A. Tay, "Performance-based optical proximity correction methodology," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 29*,* no. 1, pp. 51-64, 2010.

[57] S. X. Shi, P. Yu, and D. Z. Pan, "A unified non-rectangular device and circuit simulation model for timing and power," in *IEEE/ACM International Conference on Computer-Aided Design,* San Jose, CA, United States, 2006, pp. 423-428.

[58] K. Cao, S. Dobre, and J. Hu, "Standard cell characterization considering lithography induced variations," in *43rd Design Automation Conference,* San Francisco, CA, United States, 2006, pp. 801-804.

[59] S.-D. Kim, H. Wada, and J. 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,* vol. 17*,* no. 2, pp. 192-200, 2004.

[60] Mentor Graphics Calibre Workbench User Manual. Willsonviller, OR. [Online]. Available: http://www.mentor.com

[61] Synopsys HSPICE Application Manual [Online]. Available: http://www.synopsys.com

[62] Berkeley Predictive Technology Modeling 65nm BSIM4 Model Card for Bulk CMOS V1.0 [Online]. Available: http://www.eas.asu.edu/~ptm/

[63] F. Arnaud et al., "A functional 0.69 μm^2 embedded 6T-SRAM bit cell for 65 nm CMOS platform," in *Symposium on VLSI Technology,* Kyoto, Japan, 2003, pp. 65-66.

[64] E. Seevinck, F. J. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *IEEE Journal of Solid-State Circuits,* vol. SC-22*,* no. 5, pp. 748-754, 1987.

[65] S.-H. Teh, C.-H. Heng, and A. Tay, "Design-process integration for performance-based OPC framework," in *Proc. ACM/IEEE Design Automation Conference,* Anaheim, CA, United states, 2008, pp. 522-527.

[66] ISCAS High-Level Models [Online]. Available: http://www.eecs.umich.edu/~jhayes/iscas.restore/benchmark.html

[67] Synopsys Design Compiler Application Manual [Online]. Available: http://www.synopsys.com

[68] Cadence SOC_Encounter User Manual [Online]. Available: http://www.cadence.com

[69] Mentor Graphics Calibre OPCpro User Manual. Willsonviller, OR. [Online]. Available: http://www.mentor.com

[70] X. Wang, M. Pilloff, H. Tang, and C. Wu, "Exploiting hierarchical structure to enhance cell-based RET with localized OPC reconfiguration," in *Proc. of SPIE* vol. 5756*,* San Jose, CA, United States, 2005, pp. 361-367.

[71] D. M. Pawlowski, L. Deng, and M. D. F. Wong, "Fast and accurate OPC for standard-cell layouts," in *Proc. of the Asia and South Pacific Design Automation Conference,* Yokohama, Japan, 2007, pp. 7-12.

[72] Y. Zhang and Z. Shi, "A new method of implementing hierarchical OPC," in *Proc of International Symposium on Quality Electronic Design,* San Jose, CA, United States, 2007, pp. 788-792.

[73] J. M. Rabaey, *Digital integrated circuits : a design perspective*, 2nd ed. Upper Saddle River, N.J.: Pearson Education International, 2003.