DESIGN FOR MANUFACTURING IN IC FABRICATION: MASK COST, CIRCUIT PERFORMANCE AND CONVERGENCE

QU YIFAN

(B.Eng.,SJTU)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

NUS GRADUATE SCHOOL FOR INTEGRATIVE

SCIENCES AND ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2013

Declaration

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Qu Yifan

4 Dec 2013

Acknowledgments

Completing this PhD degree is perhaps the most challenging period of the first 26 years of my life. The best and worst moments of my doctoral journey have been shared with many people. It is a great privilege to spend four years in NUS Graduate School for Integrative Sciences and Engineering and the Department of Electrical and Computer Engineering at National University of Singapore, and its members will always remain dear to me.

I would like to express my heartfelt gratitude to Prof. Lee Tong Heng and Assoc. Prof. Arthur Tay, who are not only supervisors but also role models. Their immense knowledge and patient guidance helped me throughout my four years of research and writing of this thesis. Special thanks to my Thesis Advisory Committee, Prof. Ben M. Chen and Dr. Gan Oon Peen for their guidance and useful and practical suggestions.

I would also like to thank Assoc. Prof. Heng Chun Huat, who provided stimulating ideas and encouraging and constructive feedback, and the precious opportunity to get access to the industry tools in VLSI lab. I would not have contemplated this road if not for the generous financial support given by the NGS Scholarship, as well as the resourceful coursework supported by the intellectual and helpful lecturers. Special gratitude goes to Prof. Wang Qing-Guo, Assoc. Prof. Ong Chong Jin, Prof. Lian Yong, Dr. Yao Libin, Assoc. Prof. Xiang Cheng, Assoc. Prof. Peter Chen, Prof. Xu Yong Ping, Dr. Venkatakrishnan Venkataramanan, Assoc. Prof. Lim Kah Bin and Dr. Chui Chee Kong, who have carefully instructed me with the knowledge in the realms of control technology, circuit design and computer vision.

Members of Center for Intelligent Control also deserve my sincerest gratitude. It would be hard to complete my research without precious and friendly assistance of the members of Advanced Control Technology Lab. Special thanks go to Mdm. S. Mainavathi and Mr. Zhang Hengwei for their utmost technical and logistical support, and Dr. Teh Siew Hong, Dr. Ngo Yit Sung, Dr. Yang Geng, Mr. Ang Kar Tien, Dr. Nie Maowen, Dr. Chen Xuetao, Mr. Yu Chao, Dr. Yang Yang, Dr. Xue Zhengui, Dr. Liu Lei, Dr. Yuan Jian, Dr. Xie Jing, Mr. Qi Jing, Mr. Shi Qixian, Mr. Shen Chengyao, and all my friends in Singapore, China and other parts of the world, who are the sources of laughter and support.

I wish to thank my parents and my deceased grandmother, whose love provided my inspiration and was my driving force, and my fiancee, Miss Gu Panyu, whose love and encouragement inspired me so that I could finish this journey. I hope this work makes you proud.

Contents

Summary

The lithography process is the most critical step in the fabrication of integrated circuits (IC), accounting over a third of the total manufacturing cost. One of the key issues in the lithography process is the distortion of the printed images due to optical diffraction effect. To eliminate distortion of printed images at these advanced technology nodes, design for manufacturing (DFM) methods, such as optical proximity correction (OPC), have been implemented in the industry. Several problems exists in the current OPC techniques, such as mask cost, electrical performance and convergence issues. This thesis analyzes these problems and proposed a few novel approaches to improve OPC in terms of mask cost, circuit performance, convergence speed and run-time.

The International Technology Roadmap for Semiconductors (ITRS) identified a number of difficult OPC challenges for future technology nodes. The key challenges are to reduce OPC complexity, mask write-time and mask costs. The complexity of an OPC mask is determined by its level of fragmentation. A mask-cost-saving strategy with low fragmentation has been developed to address this issue, by using simple shapes, similar to the non-OPC schemes. The redundant sub-resolution shapes such as serifs, hammer heads and the stair-shaped edges are eliminated. The mask cost in terms of Manufacturing Electron Beam Exposure System (MEBES) file size is significantly reduced by 37% when tested on standard test chips, which can be directly translated into savings of the overall manufacturing cost, lower data volume and CPU processing time.

ITRS also highlighted that future OPC techniques should take into consideration circuit metrics such as circuit timing. This is critical since OPC edge insertion procedure may impact circuit performance. A timing-performance-aware OPC approach is developed to reduce the performance drift in circuit timing. The proposed approach optimizes post-OPC timing performance of the digital standard cells in terms of propagation delay. Simulations on benchmark circuits show up to 10% improvement compared to conventional shape-driven and electrically-driven OPC schemes. In addition, with accurate timing performance, process window could be enlarged by 88%, which means that the robustness under process variations is significantly improved.

Convergence is another important issue in OPC mask design methodology. A large number of iterations of edge perturbations are necessary in conventional OPC approaches in order to converge to the desired result. Feedback control theory is used to improve the convergence speed in the OPC iterations. A proportional-integral (PI) controller is utilized and the controller parameters are adaptively tuned with an iterative feedback tuning (IFT) algorithm for different processes. Simulation results show that the convergence speed is improved, and run-time is reduced by 80%, using various industrial standard test circuits.

Finally, for large circuits with numerous repetitive cells, a fast OPC technique is developed to accelerate the overall OPC run-time. The full layout is split into multiple single cells and OPC is conducted in parallel using lookup tables for each type of standard cell, thereby avoiding the computationally expensive full-chip OPC run-time. The average speed-up is up to 6 times when compared to conventional full chip OPC schemes.

List of Figures

List of Tables

List of Abbreviations

The following table describes the significance of various abbreviations and acronyms used throughout the thesis.

Abbreviation	**Meaning**
193nm	optical lithography using ArF laser at 193nm wavelength
λ	wavelength of the light source
ArF	argon fluoride laser
AOI	AOI gate (AND-OR-Invert)
ASIC	application-specific integrated circuit
CAR	chemically amplified resist
CD	critical dimension
CDF	cumulative distribution function
CMOS	complementary metal-oxide-semiconductor
CPU	central processing unit
DE	double exposure
DFM	design for manufacturing
DIFF	diffusion area in wafer

DoF	depth of focus
DP	double patterning
DRAM	dynamic random-access memory
DUV	deep ultraviolet
EB	edge bias
EBL	electron beam lithography
ED-OPC	electrically driven optical proximity correction
EPE	edge placement error
EPW	electrical process window
EUV	extreme ultraviolet
FF	flip-flop or latch
GDSII	Graphic Design System II
GPW	geometric process window
HDL	hardware description language
HMDS	Hexamethyldisilazane primer
Hyper-NA	$NA > 1.0$ with immersion lithography technique
IC	integrated circuit
IFT	iterative feedback tuning
INV	inverter
ISCAS	IEEE International Symposium on Circuits and Systems
ISCAS'85	ISCAS benchmark circuits (1985)
ITRS	International Roadmap for Semiconductor

k_1	a coefficient that encapsulates process-related factors
L	length of a transistor
LFD	lithography friendly design
MEBES	Manufacturing Electron Beam Exposure System
MOSFET	metal-oxide-semiconductor field-effect transistor
MP	multiple patterning
NA	numerical aperture
NAND	NAND gate (NOT-AND)
NILS	normalized image log slope
NMOS	n-channel MOSFET
NOR	NOR gate (NOT-OR)
OAI	off-axis illumination
OAI	OAI gate (OR-AND-Invert)
OPC	optical proximity correction
PDF	probability density function
PEB	post exposure back
PI	proportional-integral controller
PID	proportional-integral-derivative controller
POLY	polysilicon
PMI	process manufacturability index
PMOS	p-channel MOSFET
PSM	phase-shift mask

PVT	process, voltage and temperature
PW	process window
P & R	place and route
RAM	random-access memory
RET	resolution enhancement technique
SADP	self-aligned double patterning
SD-OPC	shape driven optical proximity correction
SPICE	Simulation Program with Integrated Circuit Emphasis
SRAF	sub-resolution assist feature
STA	static timing analysis
STD	standard deviation
TDR	timing driven retargeting
TMI	timing manufacturability index
TORSC	timing optimization ready standard cell
TPE	timing performance error
TPW	timing process window
UV	ultraviolet
W	width of a transistor
XOR	XOR gate (Exclusive OR)

Chapter 1

Introduction

1.1 Background

Advances in integrated circuits (ICs) performance over the past 30 years owe much to the progress made in lithography. Lithography is a process used to create multiple layers of circuit patterns on a chip [1]. It is currently the largest capital investment and operating cost component of a leading-edge semiconductor fabrication plant, accounting for 35% of the costs of manufacturing ICs [2–4]. It is also the key enabler and “bottleneck” controlling the device scaling, circuit performance and magnitude of integration for silicon semiconductors. This integration drives the size, weight, cost, reliability and capability of electronic systems [5, 6].

The lithography process consists of the following steps shown in Figure 1.1: vapor priming, spin coating, soft bake, alignment and exposure, post exposure bake, develop, and pattern transfer followed by resist striping [5, 7]. First, a primer called “Hexamethyldisilazane (HMDS)” is applied to the silicon wafer to improve the resist adhesion. A small quantity of resist is then dispensed onto the wafer, before the wafer is spun at high speed to deposit thin resist films uniformly. This is then followed by a soft bake step to reduce the remaining solvent concentration in the resist. In the alignment and exposure step, the resist-coated wafer has to undergo exposure to some form of radiation that will produce the pattern image on the resist. After exposure, a post exposure bake is performed to enable a chemical reaction to alter the resist solubility characteristic. The develop step utilizes chemical developers to remove exposed area of positive resist (or unexposed area of negative resist) to leave the desired mask pattern. Finally, pattern transfers such as etching, lift-off and implantation are conducted to build the micro-structures on the wafer.

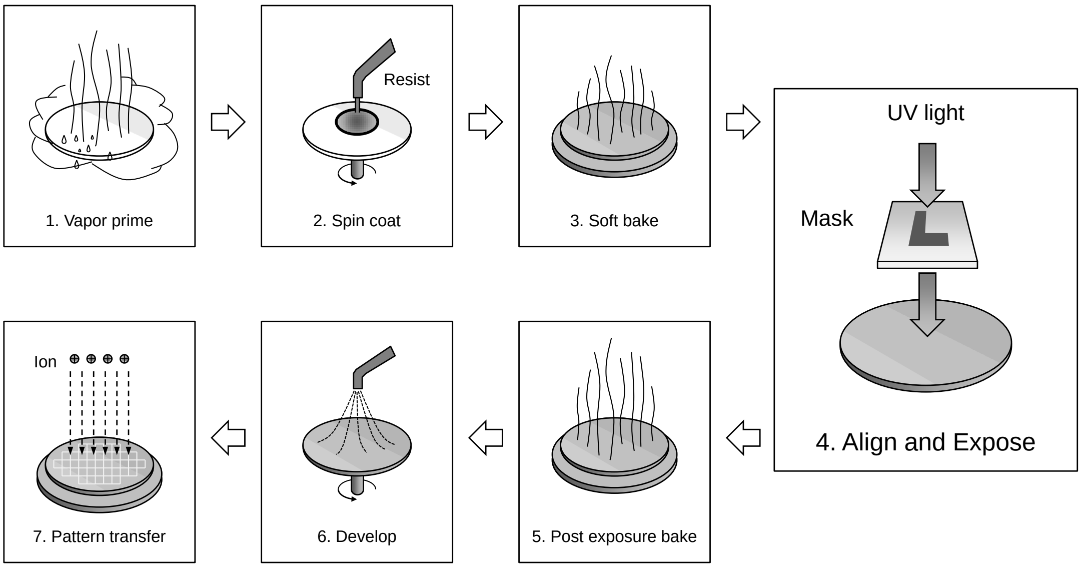


Figure 1.1: Block diagram of lithography processing steps [5]

In Figure 1.1, steps 1-3 and 5-7 are usually combined in a machine called the *track*, while the machine used to conduct step 4 is called the *aligner*. In a lithography projection process, the imaging step is always subject to degradation from diffraction which causes imperfections in the projection system. This phenomenon becomes severe when feature size scales down and it significantly obstructed printing perfect shapes onto the wafer surface.

Driven by Moore's Law [8], the number of transistors on an IC has been increasing at the pace of approximately 2 times every 18 months in the past decades. Due to the demand of putting an increasing number of transistors on the same area of a silicon substrate, the size of transistors has to be down-scaled. The scaling factor is also in line with Moore's speed: 0.7× per technology node. This trend is reported by the International Technology Roadmap for Semiconductors (ITRS) [9] as shown Table 1.1. The "critical dimension" (CD) in this table refers to the dimension of the smallest geometrical features on the semiconductor chip due to down-scaling. ITRS highlighted lithography as one of the key challenges in the next generation of technologies. As physical features of ICs shrink, lithography-induced effects, such as diffraction and optical proximity effects, become more prominent, resulting in design-for-manufacturing (DFM) issues, especially functional yield loss. It is also known as resolution limitations, as described in the following paragraphs.

Resolution in lithography is defined as the smallest feature that can be printed under adequate control. One commonly used indicator of resolution

Table 1.1: Down-scaling trend reported by International Technology Roadmap for Semiconductors [9]

Year of first product shipment	Critical dimension
2011	36
2012	32
2013	28
2015	23
2020	13

is CD (the minimum feature size in lithography). It is determined by the wavelength of the imaging light source (λ) and the numerical aperture (NA) of the projection lens according to the Rayleigh resolution criterion [10]:

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

where k_1 is a process dependent factor determined by resist capability, exposure and resist tool control, mask pattern adjustments and process control. It can be inferred from Equation (1.1) that smaller feature size can be printed by using smaller λ and larger NA. However, the optical devices are usually developed at a much lower pace than the speed at which the desired feature shrinks. Today's mainstream light source wavelength is still 193nm with argon fluoride (ArF) laser. The implementation of shorter wavelengths such as extreme ultraviolet lithography (EUV, expected to be 13.5nm) has been delayed due to immature technology for mass production [9]. In the past decades, NA has been increased from 0.16 to 1.35. Nevertheless, NA cannot continue to increase because of the depth of focus (DoF) restrictions [10]. The solution to smaller feature size is to decrease k_1, which encompasses the above-mentioned resolution limitations. This motivates the development of Resolution Enhancement Techniques (RETs).

RETs are the predominant DFM techniques in current IC design flow [11, 12]. RET approaches include Optical Proximity Correction (OPC), Off-Axis Illumination (OAI), Phase-Shift Mask (PSM), etc [7]. More recent RET approaches involve a combination of OPC, OAI and PSM. OPC is a technique to optimize mask patterns and improve fidelity of print images. OAI refers to any illumination shape that significantly reduces or eliminates the "on-axis" component of the illumination, that is, the light striking the mask at near normal incidence. Figure 1.2(a) shows two commonly used OAI sources when compared to the conventional "on-axis" source. The light is not projecting in the center area of the source. DoF can also be increased since the angle between the incident light and the mask plane is no longer perpendicular. PSM is used to overcome diffraction effects when images of neighboring parallel light beams interfere with each other. Figure 1.2(b) shows a typical phase-shift mask. A 180 degree inversion of the light beam phase can be found when transparent inversion layers (shifters) are added selectively. Resolution can thus be enhanced since less interference occurs. Among these approaches, OPC is noted as one of the key technologies enabling deep sub-wavelength IC fabriation [13]. It is also a major contributor to the mask costs and mask design turnaround time in lithography [14]. However, as feature size continues to decrease, it becomes more difficult and expensive to implement OPC [9]. Therefore, it is of immense interest to develop new techniques to reduce the cost of OPC.

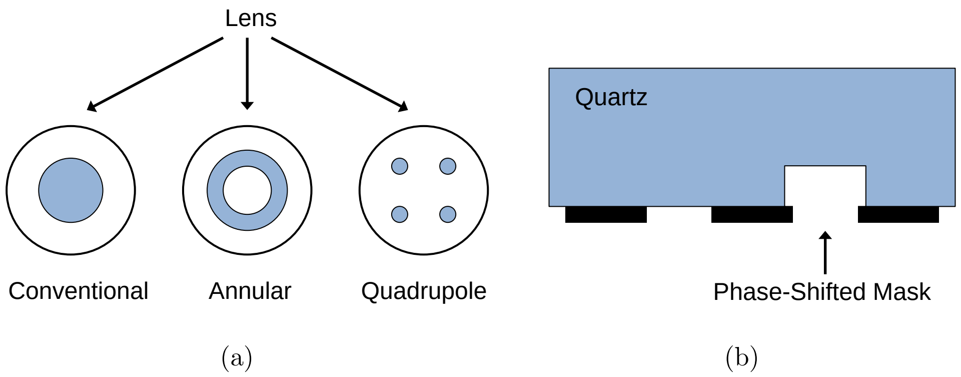


Figure 1.2: Resolution enhancement techniques: (a) OAI; (b) PSM.

OPC is an advanced mask engineering technique that is used to increase layout-to-wafer pattern fidelity. The goal of OPC is to enhance optical characteristics by making adjustments to the mask. This is accomplished by compensating mask geometry for known effects which will occur during imaging or subsequent processing [15–17]. To reiterate this more formally, as our problem statement:

Problem Statement: Given a desired geometric pattern on the wafer, find a mask design such that the final pattern remaining after the complete lithography process is as close as possible to the desired pattern.

Figure 1.3 shows an example of using optical proximity correction. If the original mask without OPC is subject to lithography process, the resulting printed image on the wafer is poor, usually far from the target shape. Improvements in shapes are found after adopting OPC. The printed images on the wafer are usually closer to the target shape.

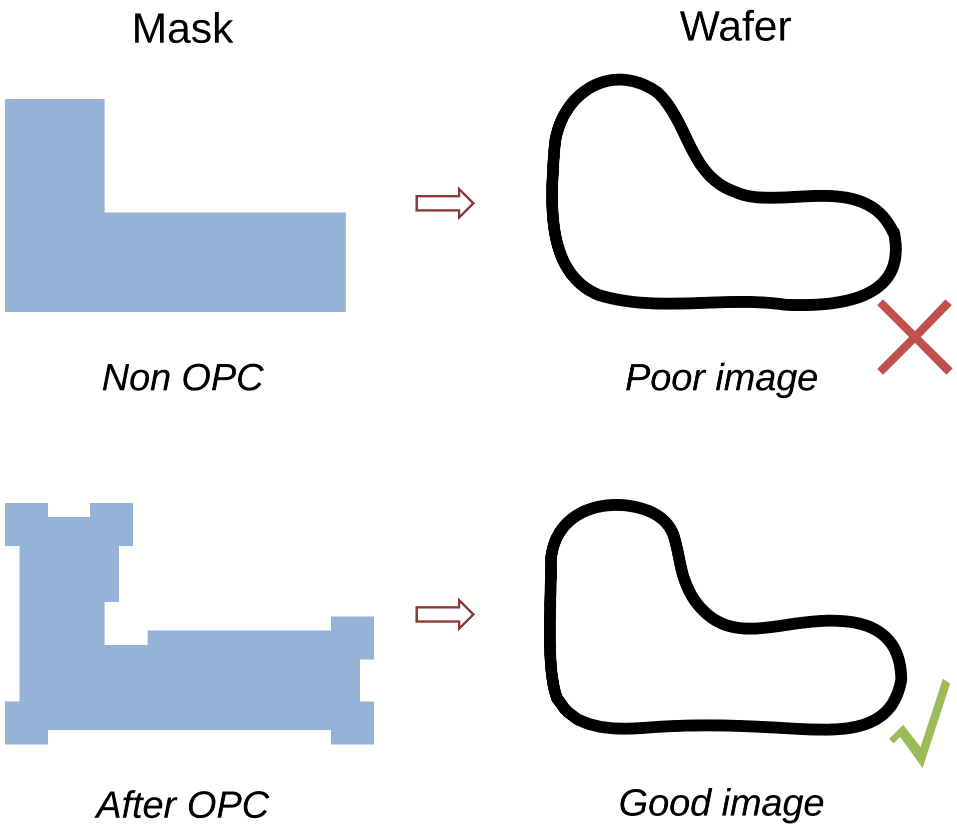


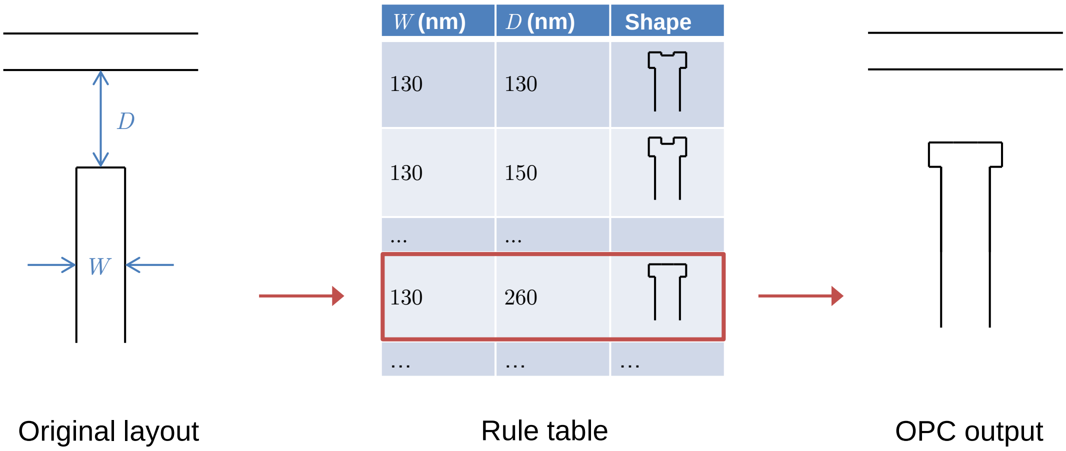
Figure 1.3: Optical proximity correction [7]

The benefits of OPC include more accurate CDs and better edge placement. Moreover, OPC enlarges process windows and improves yield for a given feature size. This allows more reliable pattern transfer at lower k_1 values. However, problems exist with the application of OPC. Masks are more complicated due to the additional vertex, fragments and sub-resolution features. [1] Run-time to generate the mask is increased owing to the number of iterations in the OPC algorithms. These problems are inevitable with the current methodologies.

[1] Fragment refers to the split short edges on the mask patterns, while vertex refers to the intersection of fragments.

1.2 Current OPC Methodologies

There are two types of OPC methodologies: rule based and model based [15]. Figure 1.4 shows an example of rule based OPC, in which feature corrections are conducted via a table look-up. The originally designed shapes are subjected to table look-ups in the rule based OPC process. These shapes are substituted according to their corresponding table entries. The overall OPC algorithm is rather simple, and run-time is an insignificant issue. However, as feature size scales downward, rule based OPC methods become incapable of dealing with mask patterns below 100nm technology node. Printed images of rule based OPC methods are no longer accurate in the state-of-the-art lithography. This motivates the development of model based OPC methods.



W (nm)	D (nm)	Shape
130	130	
130	150	
…	…	
130	260	
…	…	…

Figure 1.4: An example of rule based OPC

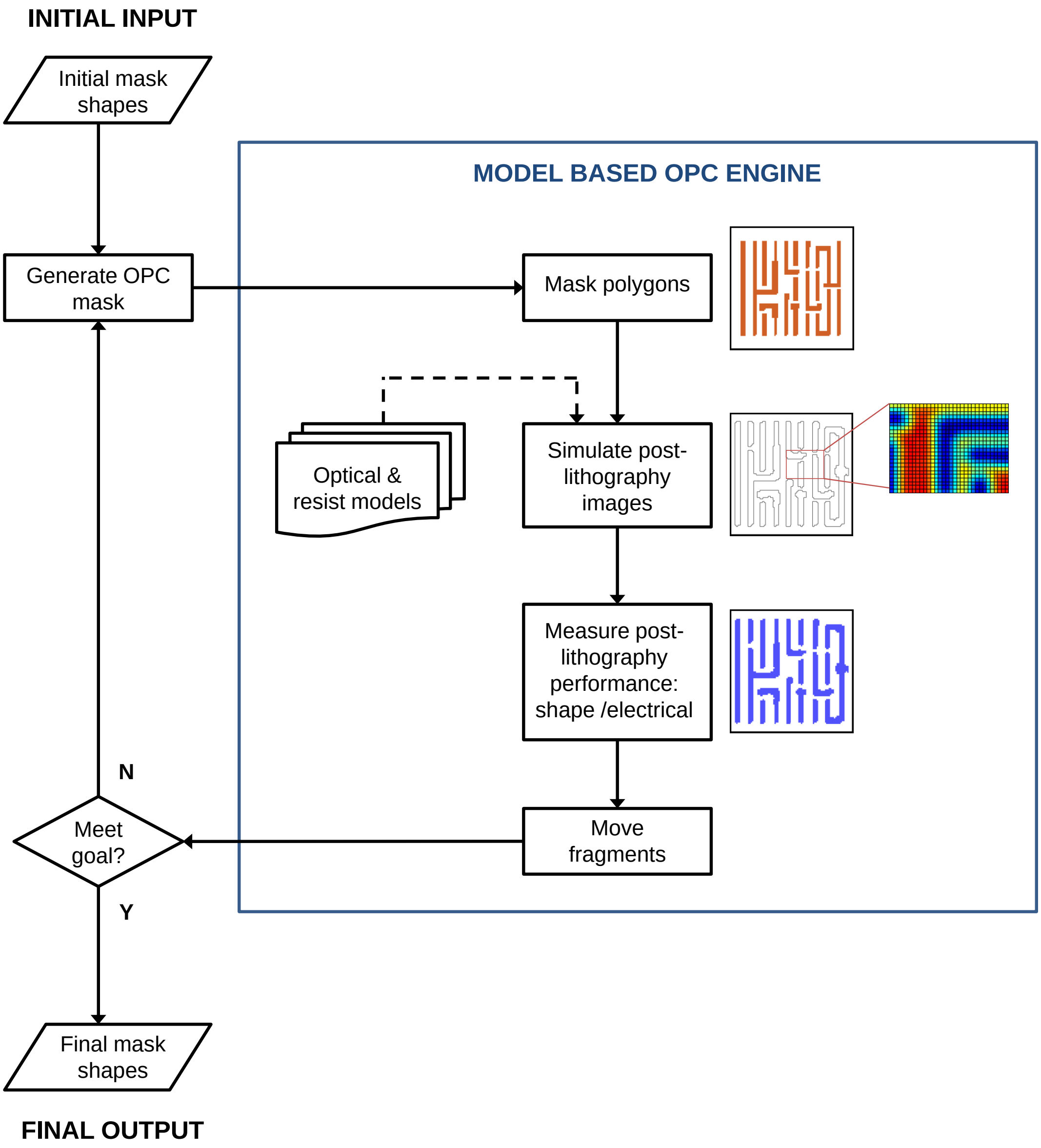


Figure 1.5: A typical flowchart of model based OPC

Model based OPC uses compact models to simulate print images dynamically and thereby move the edges on the mask to find the best solution. A typical flowchart of model based OPC is shown in Figure 1.5. The iterative model based OPC algorithm employs optical models and resist models to simulate post-lithography printed images in each loop. These printed images can be used to measure post-lithography performances such as shape and electrical metrics. The iteration stops when pre-defined criteria are met. The output in the last iteration is the final output OPC mask. Although CPU run-time of model based OPC is typically far more than that of rule based OPC, model based OPC can be applied to more complicated 2D shapes and is more accurate in terms of image fidelity. Model based OPC is the industry standard under 130nm process [18].

Most conventional model based OPC schemes are shape driven. In the model based OPC engine shown in Figure 1.5, a "measure post-lithography performance" step is conducted to measure contour errors. The most common error type is the Edge Placement Error (EPE), which is defined as the distance between the drawn edge location on the original design and the simulated edge location [18]. The objective of a shape driven OPC is to minimize the EPE so that the simulated design after exposure matches the original design closely. Figure 1.6 shows an example of shape driven OPC which tries to minimize EPE. First, a fragmentation procedure is conducted with respect to the original layout. Each edge is split into one or more fragments. In each OPC loop, EPE is calculated based on the target location and actual printed image of a fragment. Edge Bias (EB) is then derived as a function of EPE, *i.e.* $EB = f(EPE)$. A typical example of this function is: $EB = F(n) \cdot EPE$, where $F(n)$ is a function of iteration number, n. Next, the fragment is moved to a new location (a distance of EB

away from previous location). After all fragments are relocated, the new layout is subjected to a new loop. The loop does not stop until convergence reaches or stop criteria are met [18]. It is interesting to note that the above description can be formulated as a feedback problem as shown in Figure 1.7.

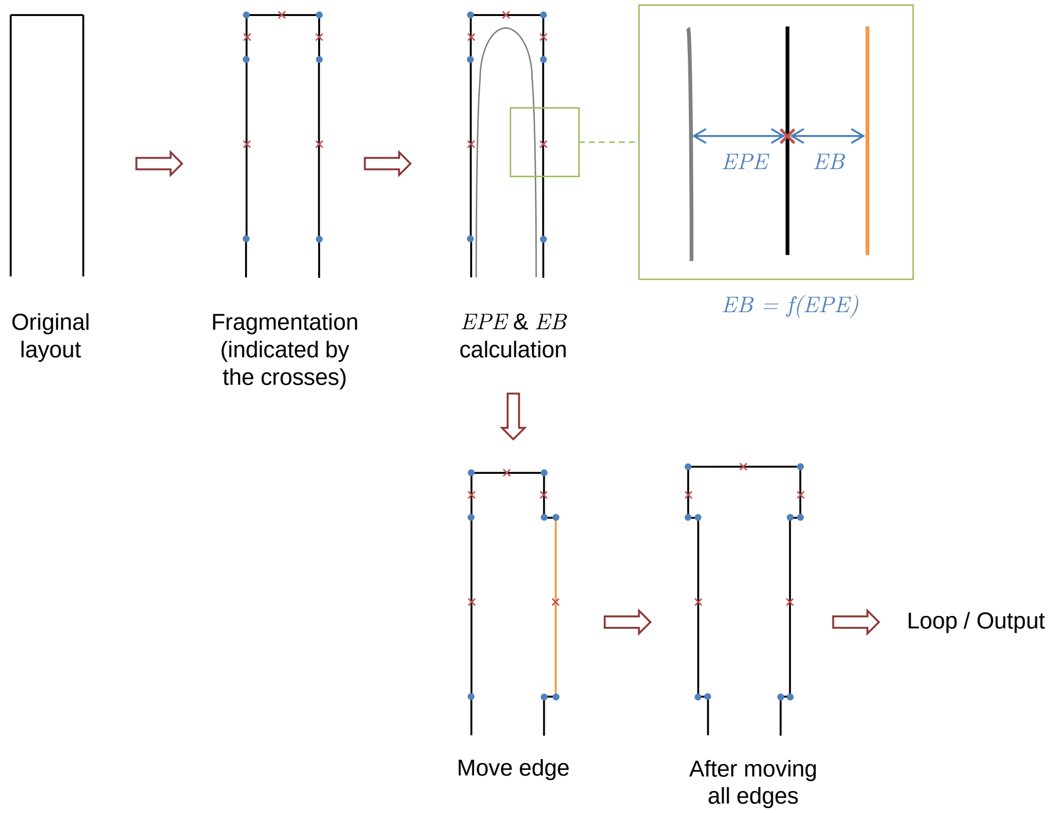


Figure 1.6: An example of model based OPC

The preprocesses of OPC, the steps before actual OPC is conducted, include Retargeting and Sub-Resolution Assist Features (SRAFs) [19, 20]. Retargeting is a process to bias the edges of the original layout before OPC is conducted, based on the knowledge in process parameters and pattern offsets. With retargeting, it usually results in faster convergence. Figure 1.8

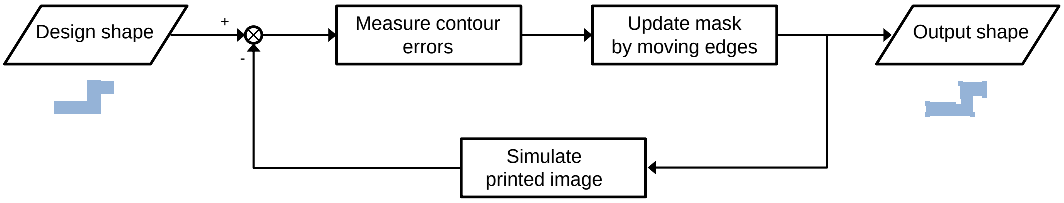


Figure 1.7: Feedback block diagram of the model based OPC flow

shows an example of retargeting. Figure 1.8(b) is the layout after retargeting, and OPC (Figure 1.8(c)) is conducted based on the post-retargeting layout.

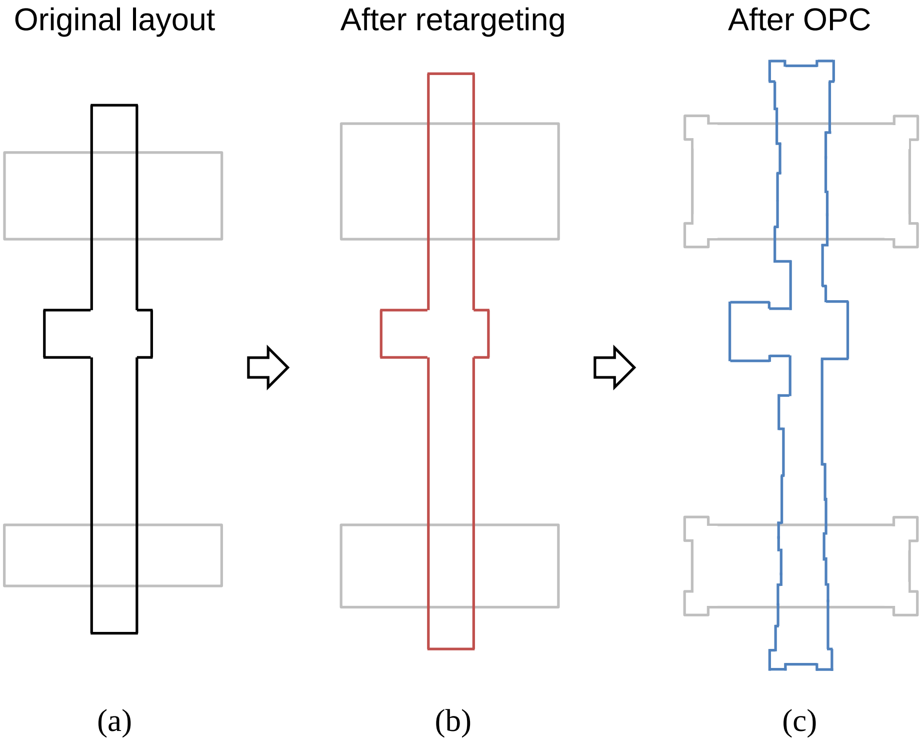


Figure 1.8: Retargeting as a preprocess of OPC

SRAF is a process to insert scattering bars and other minor patterns into the mask. The aim of SRAF is to improve contrast of light intensity and depth of focus. An SRAF is designed to improve the process margin of a resulting wafer pattern but not to be printed on the wafer. Figure 1.9 shows an example of SRAF insertion. The red polygons are SRAFs, which are part of the final mask. Although SRAFs will not appear in the printed image, the image quality can be improved.

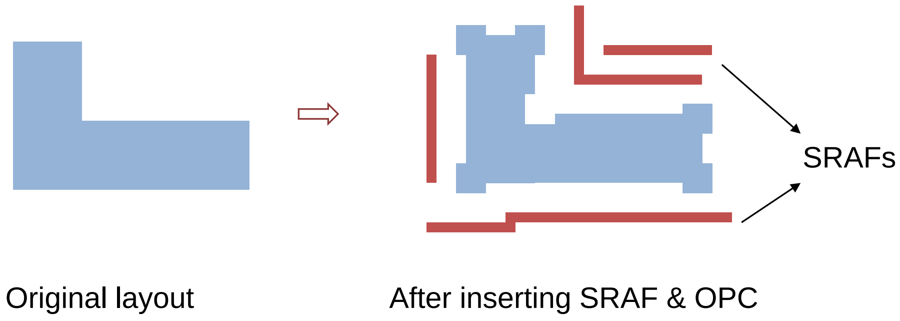


Figure 1.9: Sub-Resolution Assist Features

1.3 Motivations

Despite the apparent advantages of OPC, problems exist in its applications in the industry. Manufacturing cost largely increases due to the complex shapes in OPC, while a more desirable metric of ICs, circuit performance, is not considered in the OPC loop [14, 21, 22]. Issues of convergence and run-time are also not adequately addressed. The research gaps introduced in this section motivate this thesis to design and implement new DFM techniques.

1.3.1 Mask Cost

As reported in the lithography chapter in ITRS [9] in 2011, “reduced-cost” has become one of the key requirements of lithography. It is regarded as a “difficult challenge” to reduce complexity, write time and cost of the masks. In recent years, as OPC has been widely adopted, the complexity of the masks increased significantly. Extra vertex and extra fragments are introduced to the masks (a 4-8 times increase in fragment count [23]). The fragmentation in OPC is the main cause of the increase in extra vertex and extra fragments. The complexity of an OPC mask is determined by its level of fragmentation. High fragmentation results in short fragments and large fragment count. On the other hand, low fragmentation receives long fragments and small fragment count. Aggressiveness of OPC usually increases as fragment count increases. Figure 1.10 shows a series of OPC schemes with different levels of aggressiveness. The aggressiveness of OPC affects the mask complexity and hence the mask writing process; the mask write time is proportional to mask fragment count or vertex count [23]. The fragment counts of the three layouts in Figure 1.10 are tabulated in Table 1.2. It is clear that aggressive OPC increases mask fragment count and hence mask write time and manufacturing cost significantly.

Table 1.2: Fragment count of OPC mask in Figure 1.10

Non OPC	Moderate OPC	Aggressive OPC
6	30	67

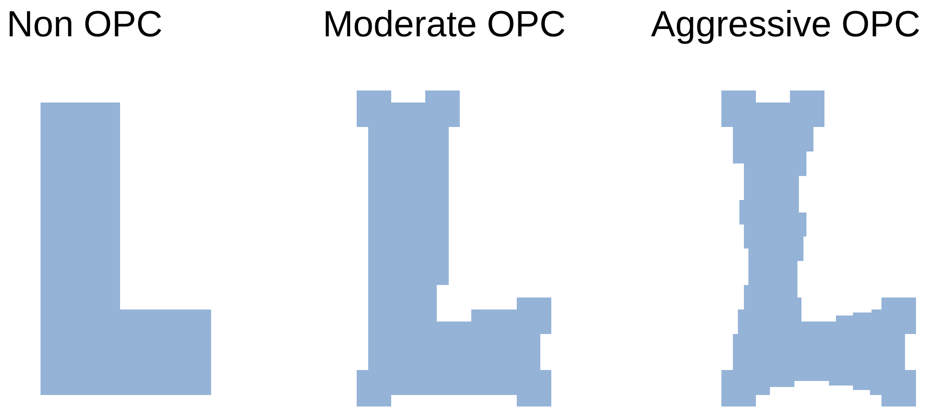


Figure 1.10: Comparison of OPC schemes with increasing aggressiveness

Apart from the mask write time, mask cost is also related with data volume [9]. Aggressive OPC drives up the Graphic Design System II (GDSII) [2] layout file size. The resulting fractured data in terms of Manufacturing Electron Beam Exposure System (MEBES) [3] file size also increases. This exponential increase in data volume is highlighted in Table 1.3. Huge mask data size does not only consume disk space, but also incurs long CPU processing time. Therefore, there is a clear need to reduce mask data volume to save disk space and run-time.

Table 1.3: Mask data volume trend reported by ITRS [9]

Year	DRAM 1/2 pitch (nm)	Mask data volume (GB)
2005	80	260
2007	65	413
2009	52	655
2011	36	1570
2013	28	2220
2015	23	2970
2020	13	Unclear

[2] GDSII stream format is the industry standard for data exchange of ICs. It is a binary file containing information of planar geometric shapes, text labels, etc.

[3] A MEBES file stores the fractured data of a mask. It is commonly supported by most mask writers [2].

The literature consists of several works on mask cost reduction [14, 22, 24, 25]. A method to reduce mask cost in the Self-Aligned Double Patterning (SADP) was introduced in [24], but it is only applicable to SADP lithography and is not feasible on other mainstream patterning methods. Gupta *et al.* [14] and Teh *et al.* [22] tried to reduce mask cost with consideration of circuit performance, but limitations in circuit performance exists (Section 1.3.1 will discuss such limitations). In [25], a regular fabric option was provided to optimize macro layout templates, but this required huge computational effort and significant layout changes.

1.3.2 Circuit Performance

According to the latest ITRS report [9], limitations in lithography hardware resolution will require design flows to more explicitly account for the impact of RETs. RET tools such as OPC must become explicitly aware of circuit metrics such as timing and power. Such awareness aligns the tools with overall product goals and enables yield enhancements, manufacturing cost reductions, and improvement in mask data preparation time. As a consequence, the tools in the application-specific integrated circuit (ASIC) design flow will have to properly plan for RET and OPC modifications downstream.

In the conventional shape driven OPC (SD-OPC) schemes, the impacts of the OPC edge insertion on the circuit performance are not taken into account during the correction. It is thus possible that an overcorrected OPC mask would just slightly outperform a moderately corrected OPC mask but at a much higher cost. Therefore, there is a need to incorporate the design intent (circuit performance) into the OPC flow to avoid the aforementioned scenario. In [14], circuit performance was incorporated into the OPC flow, where the tolerable EPE was predetermined from the timing analysis and the problem was solved as a constrained OPC insertion with geometry matching. Novel electrically driven OPC (ED-OPC) approaches based on the objective of minimizing the error in the current, rather than the EPE were also proposed in [21, 22]. Figure 1.11 illustrates the advantage of ED-OPC over SD-OPC. The first OPC scheme in the figure shows an SD-OPC while the second is an ED-OPC. ED-OPC ensures the electrical performance and its printed shapes are perhaps poor but still acceptable. On the other hand, though SD-OPC produces better printed shapes, its electrical performance is usually worse than that of ED-OPC [14, 21, 22, 26]. However, the problem of the existing ED-OPC approaches is that, the transistor drive current does not fully account for the desired circuit behavior. Instead, timing characteristics, such as propagation delay, are often a more desirable circuit behavior in digital logic gates [27]. 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 [28–31]. Specifically, the circuit performance variability under different OPC settings were analyzed off-line to quantify the different OPC dissection algorithms [31]. A unidirectional link was established to connect the OPC settings to post-OPC circuit performance but not otherwise.

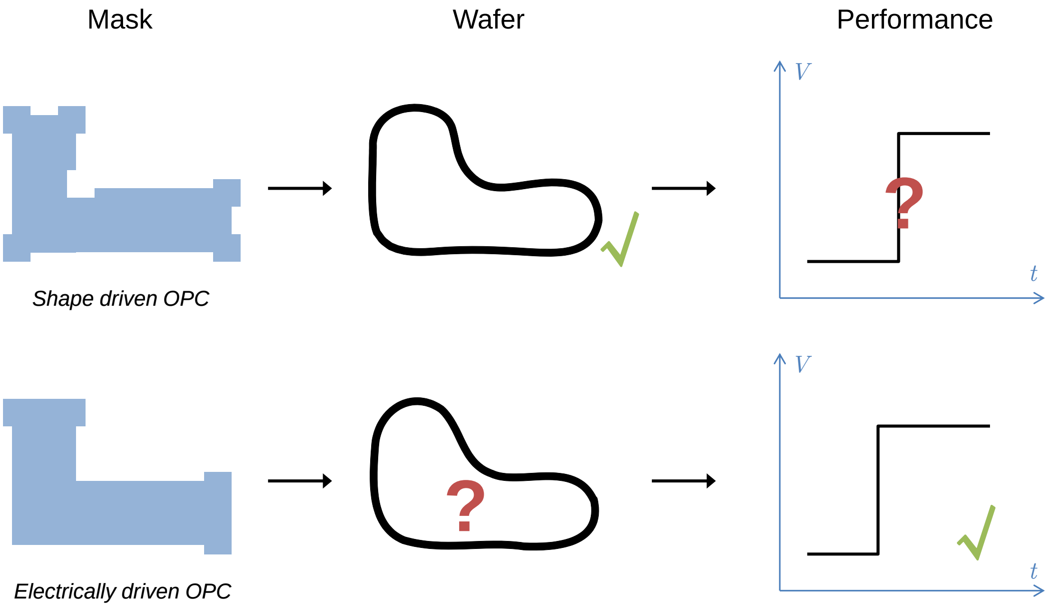


Figure 1.11: Comparison of two OPC schemes: shape driven OPC vs. electrically driven OPC

This motivates the implementation of a new technique to complete the loop by feeding back the post-OPC timing performance and develop an ED-OPC algorithm to minimize the performance variation for a given design intent. A real-time performance extraction, instead of off-line analysis, is also desirable to accelerate the OPC engine.

1.3.3 Convergence and Run-time

OPC convergence is another important issue in OPC mask design methodology. In model based OPC schemes, many iterations of the edge perturbations are necessary in order to converge to the desired result. Figure 1.12 illustrates the issue of convergence speed. The EPE step response of OPC-2 only takes 9 clock cycles to converge. However, the EPE step response of the OPC-1 is worse than OPC-2 in terms of number of iterations to converge. In some cases, oscillation and other instability issues can occur.

In IC fabrication, run-time means cost – long run-time halts the plant and causes loss in efficiency. Run-time in OPC is closely related to convergence of OPC. In model based OPC schemes, run-time is proportional to the number of loops to converge. In each loop, a computationally expensive lithography "print image" simulation is conducted. OPC with fast convergence can definitely reduce the number of iterations, and hence the length of run-time.

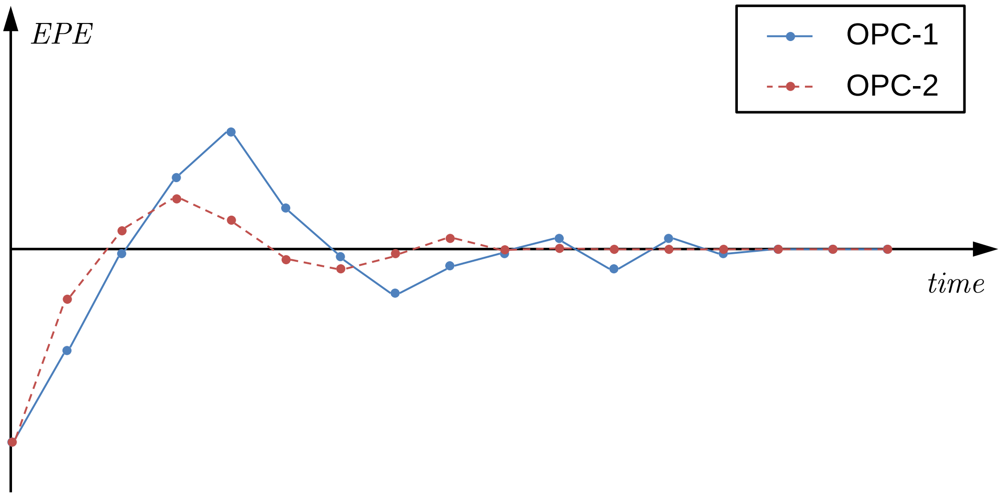


Figure 1.12: EPE step response of two OPC controllers

Therefore, there is a clear need to improve the convergence of OPC to reduce run-time and to avoid instability. Various techniques [32–35] have been proposed in the literature to improve the convergence of OPC. Figure 1.13 depicts the block diagram of a typical OPC controller. A controller is installed before the OPC plant to improve the convergence of OPC. However these techniques are primarily shape-driven OPC design methodologies and cannot be applied directly to electrically driven OPC platform. For this reason, a new technique is desirable to work with electrically driven OPC for better convergence.

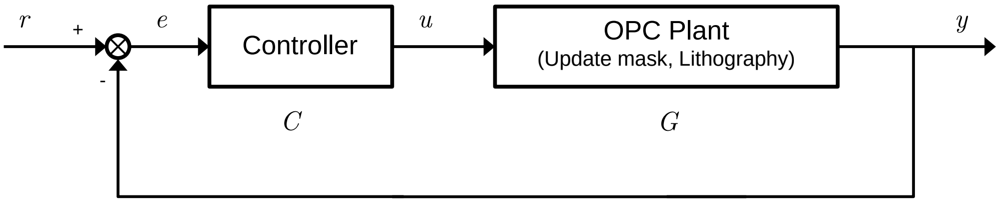


Figure 1.13: A typical OPC controller block diagram

1.4 Contributions

In this thesis, new DFM approaches are proposed to address the aforementioned problems including mask cost, timing performance and convergence. The key contributions of the thesis are listed below.

1.4.1 Timing Performance Oriented Optical Proximity Correction

The previous sections have highlighted that the transistor drive current does not fully account for the desired circuit behavior, and timing characteristics are often a more desirable circuit behavior. Transistor drive-current-oriented methods [21, 22] lead to good transistor electrical performance but not good timing performance. In this work, a timing performance oriented OPC methodology (TPO-OPC) is developed to take timing performance into account in the OPC loop. The improvements are illustrated in Figure 1.14. This OPC methodology employs the integration of a lithography simulator and SPICE simulator to estimate the timing performance of a circuit after lithography. In contrast to conventional model based OPC, the idea of shape driven OPC is converted to electrically driven OPC. Specifically, instead of transistor drive current, timing performance is selected as the metric of electrical performance, since timing performance is a more direct indicator than transistor drive current. A transistor slicing model [36] is utilized to predict timing performance of non-rectangular gates of standard cell layouts. Masks are generated via iterative knowledge-based mask correction technique.

The feasibility of the proposed TPO-OPC is demonstrated via simulations by comparing its performance against conventional OPC schemes (typical EPE-OPC approaches) built in the commercial software. An industry standard open-source standard cell library is employed as the test circuit to conduct the comparison. Simulation results show that the proposed TPO-OPC approach outperforms the conventional scheme in two aspects. First, mask size in terms of MEBES file size is reduced by 24-36%. Second, timing accuracy in terms of propagation delay is improved by 2-4%.

Further improvements in convergence can be achieved by formulating the problem into a feedback control framework. The used of the PI controller is demonstrated in this work. The PI controller parameters can be chosen based on a heuristic approach according to different optical and lithography

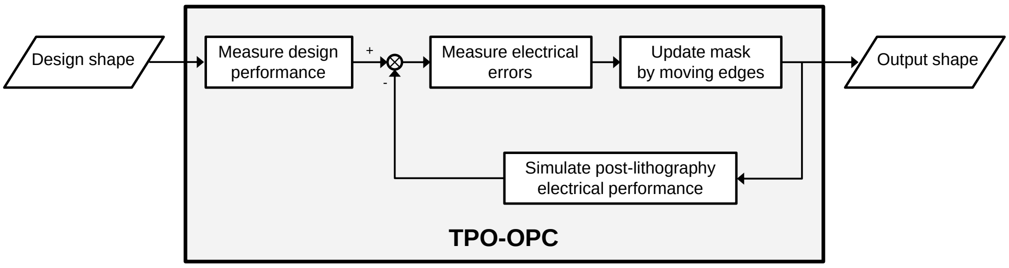

Design shape
Measure design performance
+
-
Measure electrical errors
Update mask by moving edges
Output shape
Simulate post-lithography electrical performance
TPO-OPC

Figure 1.14: Improvements of TPO-OPC from conventional model based OPC shown in Figure 1.6

settings. A major limitation of such approaches is that the choice/tuning of the PI parameters is not a straight-forward task. In addition, as various optical and lithography settings changes for different processes, the PI controller parameters have to be re-tuned properly to avoid instability and increased correction time [37]. The difficulties in obtaining an accurate, low-ordered model of the lithography process means that most PID tuning formulas in literature cannot be applied. An iterative feedback tuning (IFT) algorithm [38–40] is used to address the above issues. This approach can adaptively tune the PI parameters as process setting changes and at the same time compute the tuning parameters for the optimal performance depending on the performance index. The control block diagram with IFT is depicted in Figure 1.15. Simulation results show that the proposed method outperforms the previous methods without feedback controller: the number of iterations is reduced by 80%.

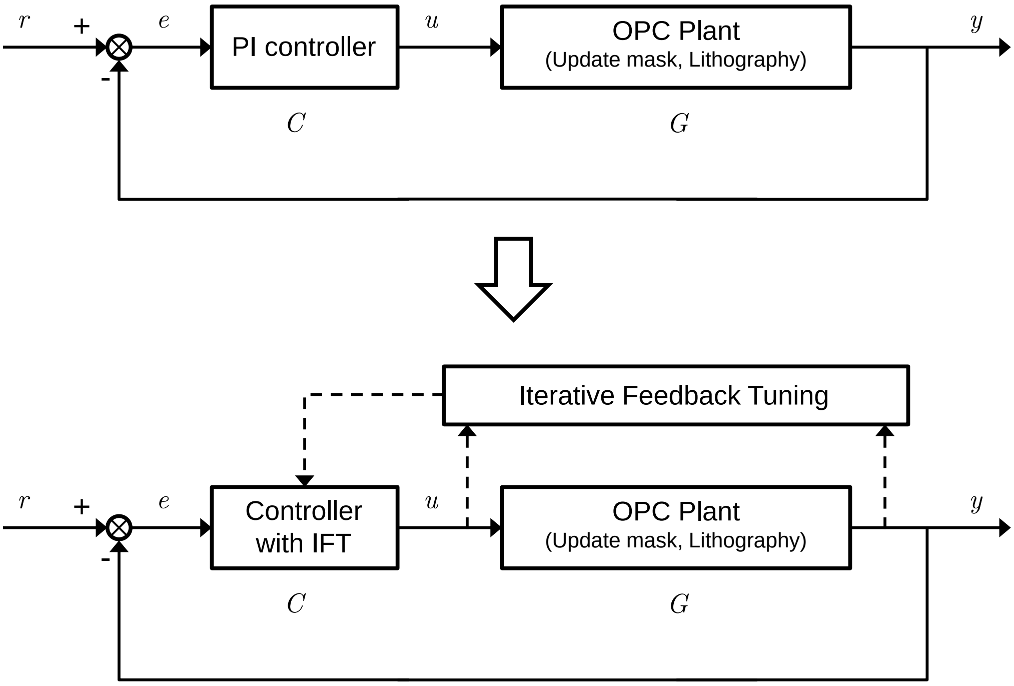

r
+
-
e
PI controller
C
u
OPC Plant
(Update mask, Lithography)
G
y
Iterative Feedback Tuning
r
+
-
e
Controller
with IFT
C
u
OPC Plant
(Update mask, Lithography)
G
y

Figure 1.15: Control block diagram of the proposed approach

1.4.2 Process Window Aware Optical Proximity Correction

A complex complementary metal-oxide-semiconductor (CMOS) gate usually consists of a number of detail timing delays with different timing behaviors. This is the main obstacle which restricts previous methods from optimizing timing performance of a standard cell in a circuit. A process window aware OPC (PWA-OPC) technique using detail timing delay as the design metrics is designed and implemented. [4] The retargeting process before applying OPC is optimized. The process window is also considered in the algorithm.

[4] Process window refers to the range of process parameters within which predefined specifications can be satisfied.

Advantages over previous methods are illustrated in the block diagram in Figure 1.16. A novel timing cost function is proposed (integrated in the timing driven retargeting part), which links timing domain electrical characteristics with shape domain mask patterns. Instead of optimizing only worst case delays, this method improves the timing performance of all cases. This implementation leads to better timing performance of the whole circuit under process variations. The main contributions of this work can be summarized as follows:

- Timing characteristics are employed as a direct metric for the proposed retargeting approach, which are not feasible in the earlier works in [21, 22] due to the complexity of circuit behavior. It was difficult to link the complex timing behavior with the printed mask pattern in the absence of the knowledge of the relationship between circuit timing performances and retargeting direction and amount. The implementation of this direct metric enables the algorithm to achieve better timing accuracy (improve by 2-5%) compared to other electrically driven OPC techniques [14, 21, 22].

- Due to the more accurate timing achieved through the proposed PWA-OPC method, the observed process window for the benchmark circuits is enlarged by 88% compared to previous methods. This directly translates to greater robustness against process variations.

- The aggressiveness of OPC can be reduced with a mask-cost-saving strategy. Therefore a 73% mask reduction in terms of mask fragment count can be observed in this work.

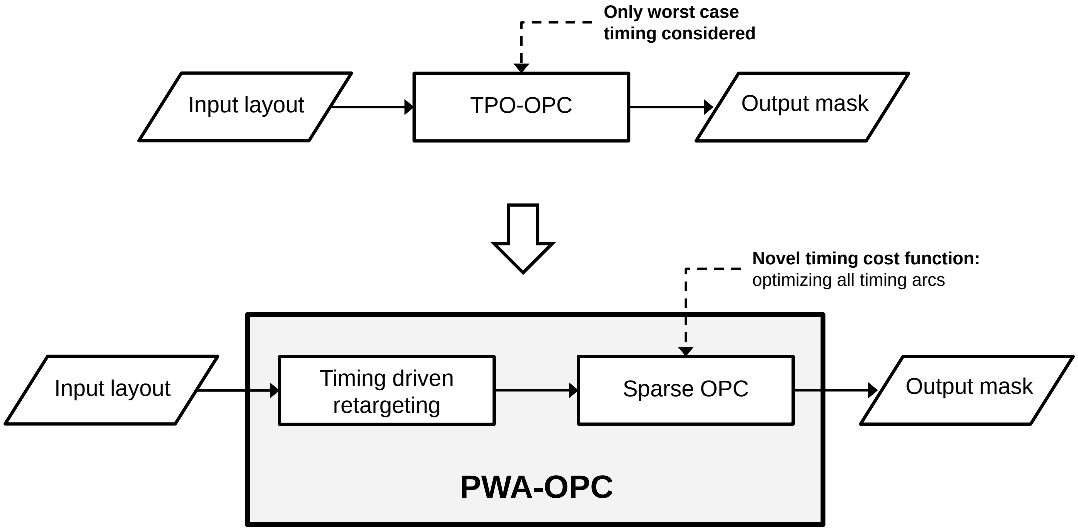

Only worst case timing considered
Input layout
TPO-OPC
Output mask
Novel timing cost function: optimizing all timing arcs
Input layout
Timing driven retargeting
Sparse OPC
Output mask
PWA-OPC

Figure 1.16: Improvements of PWA-OPC

1.4.3 Optical Proximity Corrected Mask Simplification Using Over-designed Timing Slack

Due to the variations in semiconductor manufacturing process, ICs' working voltage and temperature (PVT), standard cells are usually designed conservatively with excessive timing slack. This over-designed timing slack has not been well utilized in the past. A method to simplify OPC mask using over-designed timing slack is proposed. This method is compatible with any existing OPC schemes, including SD-OPC and ED-OPC, and the block diagram is shown in Figure 1.17. The output mask of the existing OPC schemes is treated as an intermediate mask in the proposed method. This mask is then subjected to an over-designed timing slack evaluation, before the simplification is conducted. The over-designed timing slack can be extracted from the difference between post-OPC simulations and library data. A newly proposed timing cost function enables this work to link the complex circuit behavior of timing arcs with the printed mask pattern. A transistor can be labeled with "positive shape slack" if its related timing arc has positive timing slack. The mask simplification starts with an OPCed mask which is typically generated from a commercial dense OPC engine. An iterative algorithm is applied to simplify the mask patterns with "positive shape slack" labels. In each iteration, the timing slack will be updated from simulations on the simplified new mask. The final output mask is actually the mask of minimum complexity which still ensures post-OPC timing closure. This method is implemented on Nangate 45nm Open Cell Library [41] and the major contributions can be summarized as follows:

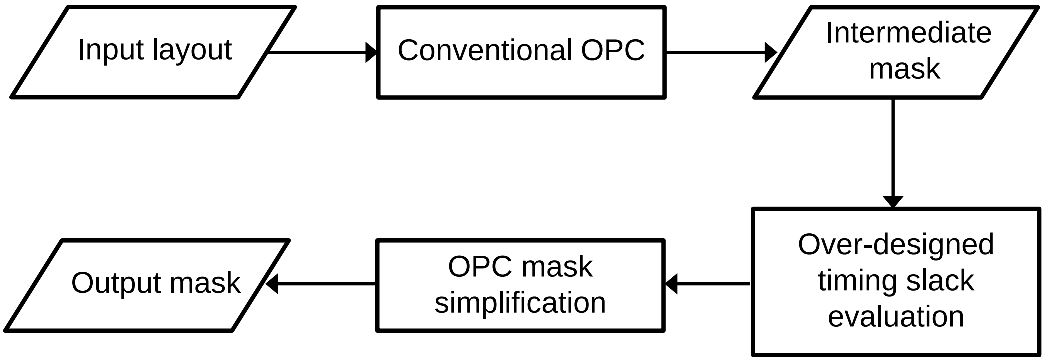


Figure 1.17: Block diagram of OPC mask simplification method

- Over-designed timing is utilized to reduce mask cost. Simulation result shows a 51% reduction in terms of polygon vertex count, which directly relates with a significant reduction in mask cost.

- A timing cost function is adopted in work to exploit the relationship between timing slack and mask patterns. The use of the transistor sensitivity vector enables this work to selectively adjust the mask patterns without violating timing restrictions. Besides, a novel mask simplification algorithm is proposed in this work to reduce the number of fragments in the OPC mask.

- This mask simplification method is compatible with the current ASIC design flow and can be implemented directly after a conventional dense OPC process. No changes will be made on the designed logic, and no more feedback (redesign signal) to the design stage will be triggered.

1.4.4 Fast Optical Proximity Correction with Timing Optimization Ready Standard Cells

As illustrated in Section 1.3, run-time in OPC is also a desirable metric. Run-time issue becomes severe when circuit size increases from one cell to hundreds of cells [42]. A fast cell-wise electrically driven OPC methodology is proposed to reduce run-time of TPO-OPC. The run-time for each cell of this OPC method (0.03sed) outperforms the fastest previous method (0.17sec) by 5 times. The overall block diagram is shown in Figure 1.18. The full layout is split into multiple single cells and TPO-OPC is conducted in parallel, for each type of standard cell. The standard cells after TPO-OPC, *i.e.* timing optimization ready standard cells (TORSCs), are stored in a lookup table. In the last step of OPC process, original layouts are substituted with the TORSCs. Run-time can be hugely saved with this method since full chip OPC is avoided.

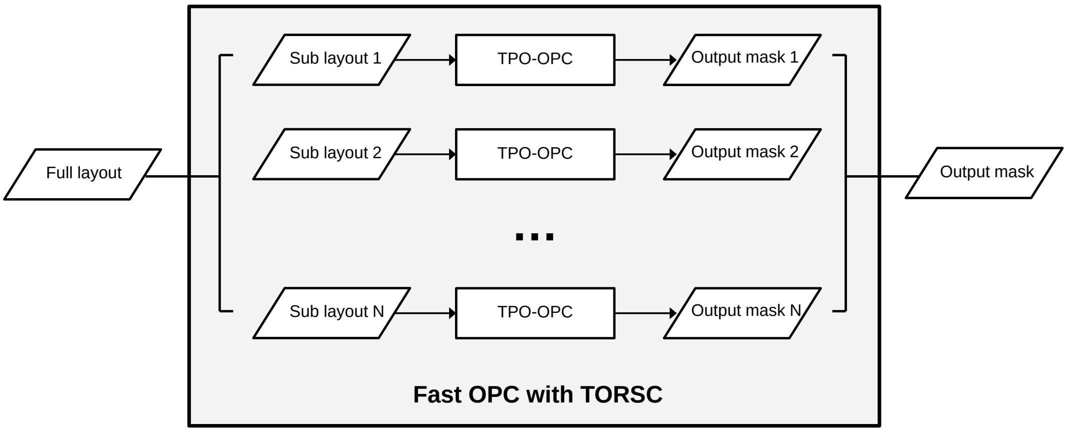


Figure 1.18: Block diagram of fast OPC with TORSC

1.5 Organization of the Thesis

This thesis is organized as follows. Chapter 2 introduces the timing performance oriented optical proximity correction (TPO-OPC) methodology. The implementation of a feedback controller and an iterative feedback tuning algorithm is also presented in this chapter to improve OPC run-time and convergence. In Chapter 3, a process window aware OPC method is proposed for both standard cell and large full chip circuits. In Chapter 4, an investigation into timing yield is conducted. Based on the discovery of the relationship between timing yield and manufacturing cost, a method to reduce mask cost using over-designed timing slack is implemented. Chapter

5 presents the development of a cell-wise OPC to reduce run-time for circuit consisting of many repeated standard cells. Finally, conclusion and future work are provided in Chapter 6.

Chapter 2

Timing Performance Oriented Optical Proximity Correction

2.1 Introduction

As timing characteristics are often a more desirable circuit behavior, a timing performance oriented OPC methodology (TPO-OPC) is proposed. TPO-OPC uses timing performance, *i.e.* propagation delay of standard cells, as the design metric. A mask-cost-saving strategy is also employed in this work to reduce the complexity of OPC masks. Simulations on 45nm technology are conducted to validate this approach. Results show that the proposed TPO-OPC outperforms conventional EPE-OPC scheme in two aspects. Firstly, mask size in terms of MEBES data volume is also reduced by 24-36%. Secondly, timing accuracy is improved by 2-4%.

A feedback controller is also implemented to improve convergence speed and stability of the OPC algorithm. Simulation results show that the proposed method outperforms the previous method without feedback controller in terms of the number of iterations. As various optical and lithography settings change for different processes, the controller parameters have to be re-tuned properly to avoid instability. To re-tune controller parameters once settings change, the use of an iterative feedback tuning (IFT) method [38–40] is adopted in this chapter. Simulation shows that, with the IFT method, TPO-OPC process achieves better convergence time: the number of iteration is reduced from about 30 to 6 for most standard cells such as INV, NAND2 and NOR3.

This chapter is organized as follows. Section 2.2 provides an overview of conventional OPC methods. The TPO-OPC methodology, including overall flow, preliminary models and algorithms, is presented in Section 2.3. Section 2.4 discusses simulation results of the proposed TPO-OPC. Section 2.5 introduces the implementation of a feedback controller and the IFT algorithm to improve the convergence of OPC algorithms. Simulation results of the feedback controller and IFT algorithm are presented in Section 2.6. Finally, conclusions are drawn in Section 2.7.

2.2 Conventional OPC Methodologies

Conventional shape driven OPC (SD-OPC) which emphasizes on printed pattern matching with designed shape often results in extensive mask corrections and higher mask cost. In particular, it is possible that an over-corrected OPC mask would just slightly outperform a moderately corrected OPC mask but at a much higher cost. There is thus motivations to incorporate the design intent (circuit performance) into the OPC flow to avoid the aforementioned scenario. The literature consists of a number of

OPC design that consider circuit performance matching [14, 21, 24, 25]. Gupta *et al.* [14] and Banerjee *et al.* [21] brought circuit performance information into their OPC flow, but they are primarily based on dense OPC schemes which still result in complicated OPC masks. In [25], a regular fabric option is provided to optimize macro layout templates, but this requires huge computational effort – all neighboring cells are combined into macro blocks under a restricted design rule – and results in significant layout changes. Zhang *et al.* [24] proposed an OPC method which considers circuit performance of the 1D cells in self-aligned double patterning lithography [43, 44], specifically for random-access memory (RAM) manufacturing, and hence cannot be applied directly in dominant patterning techniques for mainstream logic circuits.

In model based OPC, extensive computational time is required to iterate the desired mask during the OPC process [15]. The simulation for lithography process models is through complex 2D systems that require huge computation. This further increases run-time for each iteration. The total run-time of an OPC process closely relates with the design for manufacturing (DFM) time line and manufacturing cost. In [22], a simple fixed-step-size method is used to iterate the mask patterns. The process of this method usually requires more than 50 iterations. A more efficient approach to achieve faster convergence is desirable. The applications of classical feedback control theory [45] to improve the shape driven OPC correction convergence were first introduced by Painter *et al.* [37]. Su *et al.* [46] further improves the convergence issue in the feedback controller for OPC. However, it has not found widespread use in the

OPC community, since these controllers were only designed for shape driven OPC methods. We attempt to address these limitations here. Extension of these approaches to improve the OPC correction convergence for electrically driven OPC is an indirect and more complicated problem due to the required generation of the various process indexes.

Preliminary implementations of controllers for OPC are with fixed controllers. These controllers are based on fixed step sizes (usually multiples of manufacturing gird such as 1nm) in their iterative algorithm. The controller parameters are selected with a heuristic method. However, different lithography settings (wavelength, illumination type, numerical aperture, photoresist threshold etc.) maybe applied for different processes. Therefore, a fixed controller has to be re-tuned for each set of settings. The advantage of this approach is that it does not require prior modeling of the plant. The tuning could be applied online while the system is running in a closed loop. The algorithm behind this tuning method is based on an iterative Newton-like method, which targets at minimizing a predefined cost function. The tuning can start with a given set of stable controller parameters.

2.3 Timing Performance Oriented OPC

2.3.1 Overall Flow

A typical model-based EPE-OPC flowchart is shown in Figure 1.6. It minimizes the local geometry distortions: the displacement error between the designed edge and that of printed results [15], *i.e.* edge placement error (EPE). Figure 1.14 shows the flowchart of the proposed TPO-OPC method. TPO-OPC starts from the designed layout and circuit. The simulated designed timing performance, T_{design}, is set as the input to the OPC engine. The OPC engine generates a mask according to the TPE value of each iteration, where TPE refers to timing performance error and is defined as follows:

$$TPE = \frac{T_{post-lithography} - T_{design}}{T_{design}} \cdot 100\%. \tag{2.1}$$

$T_{post-lithography}$ is the timing performance, which is simulated from the printed image of the intermediate mask. It is the feedback signal in the TPO-OPC loop. This process iteratively reduces absolute value of TPE until this value in the ith iteration drops below a predefined threshold value, *i.e.* $|TPE(i)| \leq TPE\_TH$. The mask at the last iteration is the final TPO-OPC output.

2.3.2 Timing Performance Extraction

The non-ideal shapes after the lithography process are also known as “printed images”. In the past, it is difficult to simulate the electrical performance of non-ideally shaped transistors. Poppe *et al.* proposed a method [36] to simulate the electrical performance of these transistors. It is conducted using a transistor slicing method as shown in Figure 2.1. The overlapping area of polysilicon layer and diffusion layer – also known as a transistor's channel region – is horizontally split into slices with different sizes. These slices are treated as parallel transistors. The total current of a whole non-ideally shaped transistor can be calculated by summing up the current of each slice:

$$I_{total} = \sum_{k=1}^{m} I_k = \sum_{k=1}^{m} I_{wide} L_k \times \frac{W_k}{W_{wide}}, \tag{2.2}$$

where m is the number of slices, L_k is the length of slice k, W_k is the width of slice k. It has been shown that both drive and leakage current (I_{on} and I_{off}) are linearly proportional to W [22, 36]. SPICE models such as PTM HP/LP are used to simulate the total current of the slices [47, 48]. This SPICE simulation usually requires long run-time. Therefore, lookup tables for I_{on} and I_{off} can be built to speed up the calculation. [1] The total current of a non-ideally shaped transistor can be used to capture the equivalent channel lengths, $L_{eq,I_{on}}$ and $L_{eq,I_{off}}$. These equivalent channel lengths can be used to extract the post-lithography timing performance, $T_{post-lithography}$.

TPO-OPC differs from device performance-based OPC method (DPB-OPC in [22]) mainly in the aim of OPC. DPB-OPC targets at matching drive and leakage currents (I_{on} or I_{off}) of the transistors. Instead, TPO-OPC targets at matching timing performance, since timing performance is a more direct indicator than drive and leakage currents. IC designers focus on timing performance rather than device currents [27]. Simulations of DPB-OPC show that, even when the device currents are accurate, the timing performance still deviates up to 4% for the simplest

[1] In (2.2), W_{wide} and I_{wide} refer to the data from this lookup table.

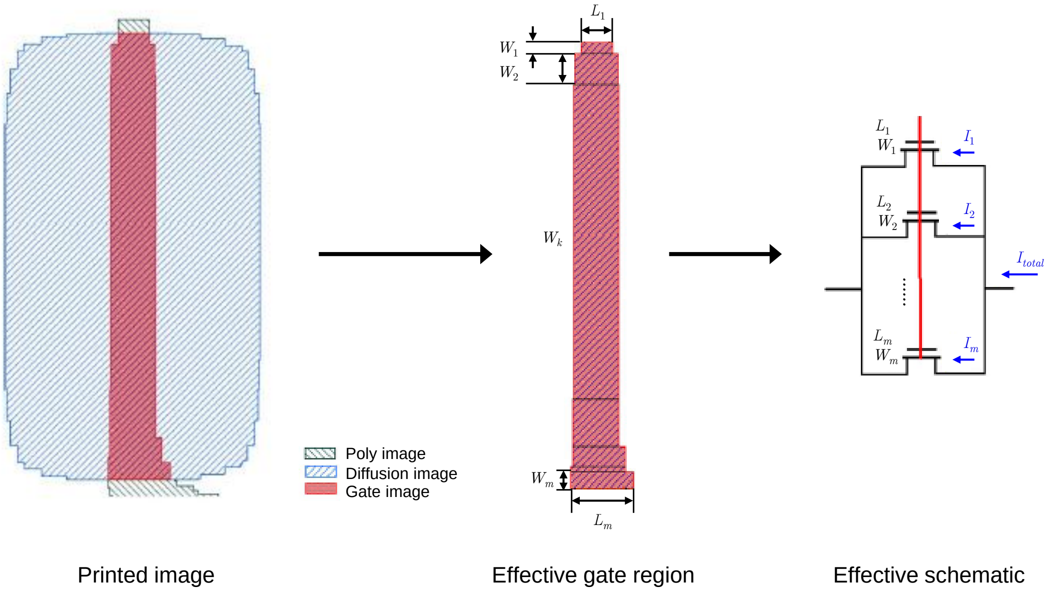


Figure 2.1: Illustration of non-rectangular gate slicing method

inverter [22]. This deviation becomes severe under process variations. Accurate timing is regarded as one key solution to enhance the robustness of the circuits under process variations [49, 50]. Worst case propagation delays (t_{pHL} and t_{pLH}) are widely used in digital integrated circuit as a pointer to the timing performance of a combinational logic [27]. Due to the importance of worst case propagation delay in application, it is selected as the designator of timing performance to best represent timing performance in this work.

2.3.3 Mask Generation Algorithm

The TPO-OPC mask generator's target is to minimize the deviation between post-lithography timing performances and designed timing performances. In particular, the timing performance error TPE is defined as follows:

$$\begin{aligned} TPE_{HL} &= \frac{t_{pHL,post-lithography} - t_{pHL,design}}{t_{pHL,design}} \cdot 100\% \\ TPE_{LH} &= \frac{t_{pLH,post-lithography} - t_{pLH,design}}{t_{pLH,design}} \cdot 100\% \end{aligned} \tag{2.3}$$

An iterative algorithm is proposed to conduct TPO-OPC, using the values of TPE, and is summarized in Figure 2.2. In this algorithm, Step 12 runs on a lithography simulator (Calibre Workbench [18]), and Step 13 utilizes a SPICE simulator (HSpice [47]). The subroutines, “Faster” and “Slower”, are transistor re-sizing functions which are introduced in the following paragraphs.

Rabaey *et al.* [27] proved that t_{pHL} is dominated by the sizes of NMOS transistors and t_{pLH} is dominated by the sizes of PMOS transistors. Hence, W (width of a transistor) and L (length of a transistor) of NMOS transistors can be adjusted to obtain larger or smaller t_{pHL}. Similar effect on t_{pLH} happens when W and L of a PMOS transistor is adjusted. Transistor re-sizing subroutines (“Faster” and “Slower”) in Figure 2.2 are based on this principle. The amount to adjust is related with the value of feedback signal TPE. When positive TPE occurs, L can be increased and W can be decreased. When negative TPE occurs, W can be increased and L can be decreased. The difference between adjusting L and W is the sensitivity. Adjusting L is more sensitive: little amount of adjustment incurs large change in TPE. On the contrary, adjusting W is less sensitive. Therefore, when large $|TPE|$ (usually greater than 10%) occurs, adjusting L is preferred. When small $|TPE|$ occurs, adjusting W is preferred. Adjusting L

and W are known as “coarse tuning” and “fine tuning”, respectively. Figure 2.3 illustrates the detail adjusting method. The polysilicon and diffusion areas of the transistors are adjusted to change t_{pHL} and t_{pLH}.

```
input : Designed layout, designed netlist, technology library, designed
        timing performance
output: TPO-OPC mask
Initialize TPE; $i \leftarrow 0$;
while $|TPE(i)| > TPE\_TH$ do
    if $TPE_{HL}(i) > 0$ then
        Faster(NMOS);
    else
        Slower(NMOS);
    if $TPE_{LH}(i) > 0$ then
        Faster(PMOS);
    else
        Slower(PMOS);
    mask $\leftarrow$ Layout($i$);
    lithography simulation;
    $L_{eq}$ extraction and SPICE simulation;
    $T_{post-lithography} \leftarrow$ post-lithography timing;
    $TPE(i+1) \leftarrow \frac{T_{post-lithography} - T_{design}}{T_{design}} \cdot 100\%$;
    $i \leftarrow i + 1$;
```

Figure 2.2: TPO-OPC algorithm

2.4 Simulation Results

Simulations are conducted based on an open source cell library for academic usage: Nangate 45nm Open Cell Library [41]. Mentor Graphics Calibre Workbench [18] is utilized as lithography simulator and EPE-OPC mask generator. 193nm light source and 1.35 hyper-NA immersion lithography are adopted. The TPO-OPC engine are written in Perl scripts and run on a Linux workstation. 5 combinational logic standard cell test circuits from [41]

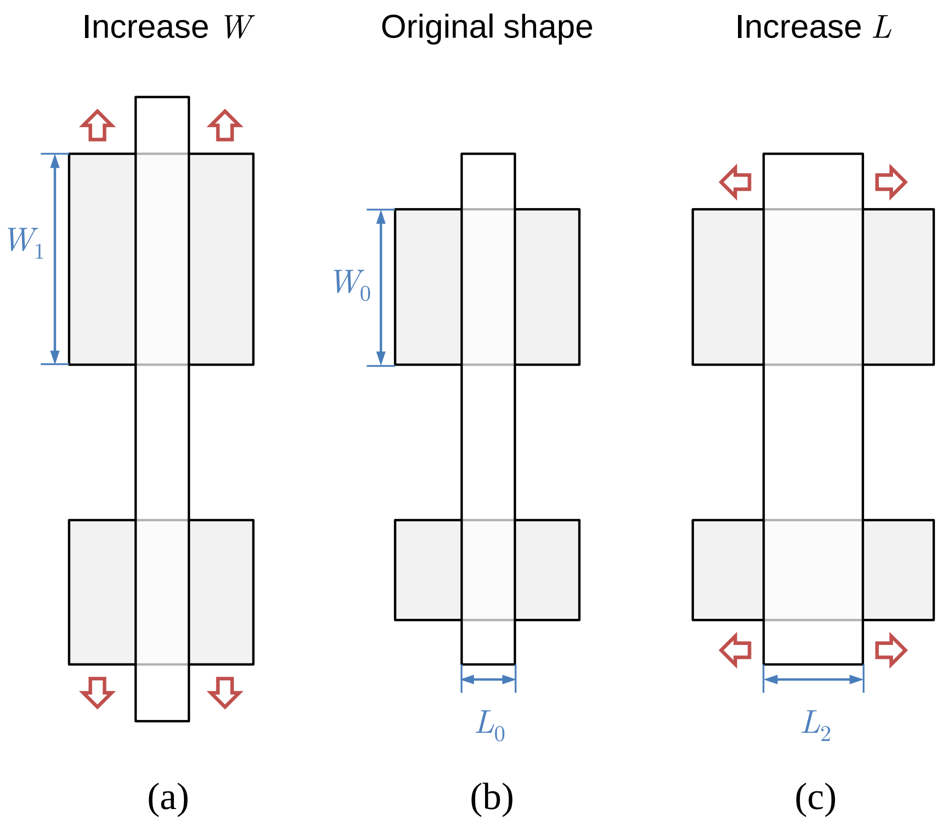


Figure 2.3: Transistor resizing: (a) width increase from W_0 to W_1, (b) original shape, (c) length increase from L_0 to L_2

are selected: INV, NAND2, NOR3, AOI211, and XOR2 (transistor numbers: 2, 4, 6, 8, & 10 respectively). To demonstrate the feasibility and effective of the proposed method, TPO-OPC and a conventional EPE-OPC are applied to these test cells. Details of the implementation of TPO-OPC is discussed in Section 2.3. The conventional EPE-OPC is conducted using Calibre Workbench, with a pre-established OPC recipe [22].

Table 2.1 and Table 2.2 show the comparison among the timing performances of designed values, TPO-OPC results, and conventional EPE-OPC results. Timing results are normalized with respect to t_{pHL} of

INV. The two tables show that deviation of t_{pHL} drops from 5.39% (EPE-OPC) to 1.08% (TPO-OPC), and deviation of t_{pLH} drops from 5.97% (EPE-OPC) to 1.25% (TPO-OPC). As compared to transistor drive current oriented methods in Refs. [21, 22], a 2-4% improvement is also achieved.

Table 2.1: Falling edge (t_{pHL}) timing performance comparison (normalized t_{pHL}, with respect to INV design t_{pHL})

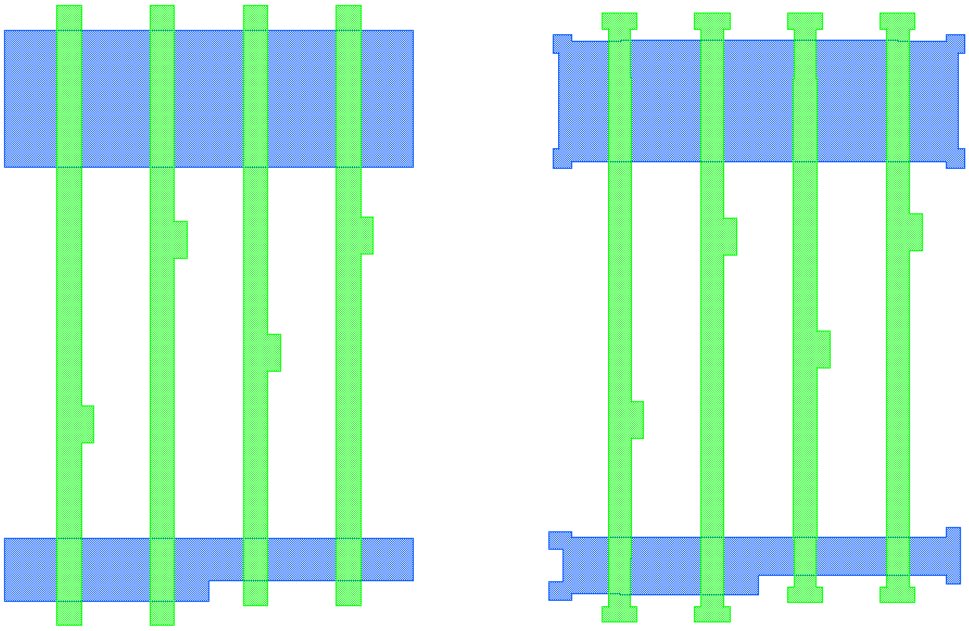
Cell	Design	TPO	|Deviation|	EPE	|Deviation|
INV	1.00	0.99	1.18%	0.95	4.60%
NAND2	1.24	1.22	1.50%	1.24	0.39%
NOR3	1.29	1.29	0.17%	1.40	8.57%
AOI211	1.46	1.45	0.79%	1.60	9.51%
XOR2	2.17	2.14	1.77%	2.09	3.89%
Average |Deviation|			**1.08%**		**5.39%**

Table 2.2: Rising edge t_{pLH} timing performance comparison (normalized t_{pLH}, with respect to INV design t_{pHL})

Cell	Design	TPO	|Deviation|	EPE	|Deviation|
INV	1.95	1.95	0.02%	1.78	8.36%
NAND2	2.16	2.19	1.33%	2.05	5.24%
NOR3	3.88	3.96	1.95%	3.71	4.34%
AOI211	4.41	4.48	1.62%	4.12	6.61%
XOR2	3.53	3.57	1.35%	3.34	5.29%
Average |Deviation|			**1.25%**		**5.97%**

Figure 2.4 shows the two masks of cell AOI211 generated by TPO-OPC and EPE-OPC respectively. The masks of TPO-OPC are less complex. Mask of TPO-OPC have no extra features at line-ends and around corners, while mask of EPE-OPC adds on extra shapes. Therefore, the mask complexity of TPO-OPC is significantly smaller. MEBES file sizes (a useful metric to evaluate mask cost) are extracted to compare the mask cost [14].

Table 2.3 shows the comparison of MEBES file sizes between the two approaches. Diffusion and polysilicon masks are reduced by 35.96% and 24.21% respectively from EPE-OPC to TPO-OPC.



(a) (b)

Figure 2.4: Mask comparison of AOI211: (a) TPO-OPC; (b) EPE-OPC. (Only polysilicon and diffusion layers are displayed.)

Table 2.3: MEBES mask size comparison (unit: Byte)

	Diffusion		Polysilicon	
Cell	TPO	EPE	TPO	EPE
INV	114688	129024	153600	153600
NAND2	114688	165888	157696	233472
NOR3	114688	172032	208896	309248
AOI211	114688	229376	309248	432128
XOR2	114688	251904	315392	436224
TPO/EPE-1	**-35.96%**		**-24.21%**	

2.5 Application of Feedback Control to Improve Convergence

In the preliminary implementation of TPO-OPC in Section 2.3, a fixed-step-size method is employed. This implementation results in slow convergence ($20 - 50$ iterations). This slow convergence has also been found in other previous OPC schemes [10, 37, 46, 51]. Since run-time reduction is a desirable target in OPC, this section demonstrates the application of feedback control to improve the convergence speed of the TPO-OPC mask design.

2.5.1 PI Controller

Figure 2.5 shows the control block diagram for automating the OPC mask design. The reference signal, r, is the desired circuit performance, *i.e.* t_{pHL} or t_{pLH}. The control signal or plant input, u, is the fragment shift in transistor width, ΔW. For the proposed PI controller, the magnitude of control signal of a certain fragment/edge of a transistor `frag(i)` in the n_{th} iteration is given by:

$$u(n) = K_P \cdot e(n) + K_I \cdot \sum_{j=0}^{n-1} e(j) = \Delta W(\texttt{frag(i)}), \qquad (2.4)$$

where $e(j)$ is the error signal in the j_{th} loop, K_P is the proportional control parameter, K_I is the integral control parameter, and n is time, which is also known as the “iteration number”. All control parameters are constants that can be tuned through off-line simulations for better convergence. Simulations are conducted to validate the feasibility and effectiveness of this PI controller and is discussed in Section 2.6.

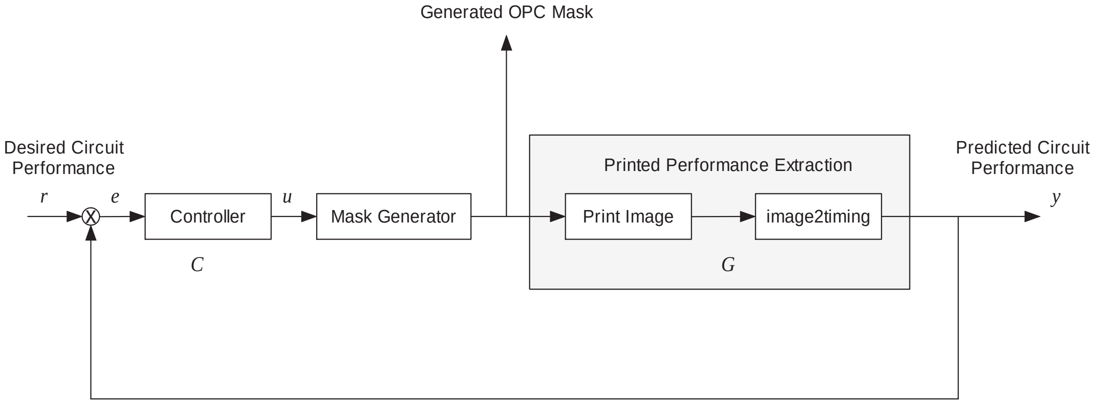

Generated OPC Mask
Desired Circuit Performance
r
e
Controller
C
u
Mask Generator
Printed Performance Extraction
Print Image
image2timing
G
Predicted Circuit Performance
y

Figure 2.5: Control block diagram in TPO-OPC

2.5.2 Iterative Feedback Tuning

It is clear that the application of feedback control theory has potential benefits in speeding up the convergence time of the OPC mask. However, in many manufacturing environment, different exposure tools, equipments and materials are used, resulting in different optical, photoresist models, just to name a few. The whole process is also complicated by the circuit performance extraction technique. In addition, it is also clear that proper tuning of the PI controllers is not easy. In the complicated lithography manufacturing systems, modeling of the optical systems is a rather difficult task. Tuning controllers for such systems requires extensive computer computation under human guidance.

To address the controller tuning and model-variation issues, the use of an adaptive algorithm that can re-tune itself under different settings is proposed.

The Iterative Feedback Tuning (IFT) algorithm is one such algorithm. It also has the added advantage that a model of the system is not required. The IFT algorithm acts as a supervisory controller which analyzed the difference of the timing performance between the desired and current mask iteration after each iteration run. The PI controller is re-tuned after each run based on an optimization function. The IFT algorithm is described elsewhere in the literature [38–40]. Details of IFT algorithm can be found in Appendix A.

2.6 Simulation Results of PI Controllers

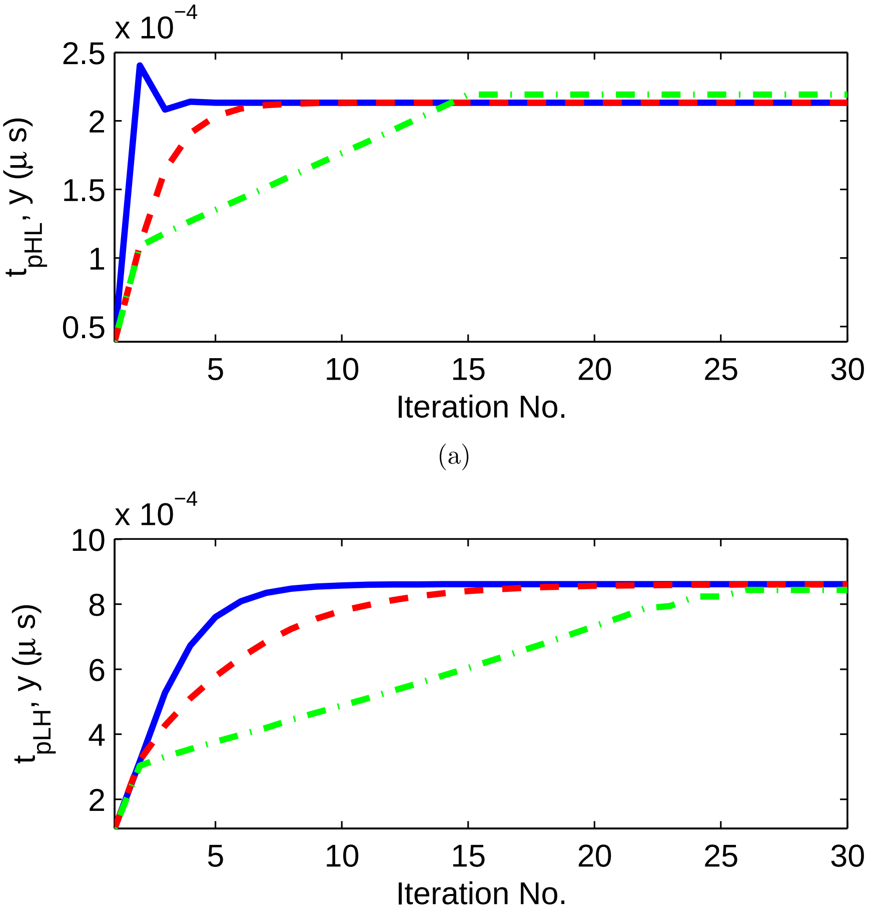
Simulations are conducted to validate the feasibility and effectiveness of the proposed PI controller and IFT tuning method. The simulations settings are the same as in Section 2.4.

2.6.1 Basic PI Controller

Simulations are conducted to evaluate the convergence of the PI controllers with different schemes. Two sets of PI controller parameters are illustrated in Figure 2.6: $K_{P1} = 1.0 \times 10^9, K_{I1} = 1.0 \times 10^6$ and $K_{P2} = 0.5 \times 10^9, K_{I2} = 0.5 \times 10^6$. [2] The fixed-step-size method used in Section 2.3 is included for comparison. Figure 2.6 shows the result of the simulation on a standard cell NOR3. With the implementation of PI controllers, faster convergence than fixed-step-size method is observed in both NMOS and PMOS sites. The first set (K_{P1}, K_{P1}) achieves a better convergence compared to the second set

[2] The initial controller parameters are calculated based on the responses of fixed-step-size method. In the fixed-step-size method, an x_0 nanometer change in transistor width averagely leads to a change of T_x nanosecond changes in timing performance. Therefore, K_P can be chosen as $k \cdot x_0/T_x$, where k is a positive number.

(K_{P2}, K_{P2}) the other. Besides, with same controller parameter values, PMOS sites have slower response speed.



(a)

(b)

Figure 2.6: Comparison of step response of cell NOR3: (a) NMOS site, (b) PMOS site. Solid line: $K_{P1} = 1.0 \times 10^9, K_{I1} = 1.0 \times 10^6$; dashed line: $K_{P2} = 0.5 \times 10^9, K_{I2} = 0.5 \times 10^6$; dash-dotted line: fixed-step-size. An initial width of 50nm is set for all transistors, so the initial output y of each run remains the same.

2.6.2 Controlling OPC plant using other PID algorithms

We have previously conducted extensive study on PID algorithms and its applications in semiconductor manufacturing [52-54]. In this section, two additional type of PID controllers are applied to the TPO-OPC plant. The performances of these controllers are compared to the PI controller proposed in Section 2.5. A pure proportional controller (P controller) and a proportional-integral-derivative controller (PID controller) are implemented using similar architectures. The control signals in the n_{th} iteration, $u(n)$, are given by:

$$u_P(n) = K_P \cdot e(n), \tag{2.5}$$

$$u_{PID}(n) = K_P \cdot e(n) + K_I \cdot \sum_{j=0}^{n-1} e(j) + K_D \cdot [e(n) - e(n-1)], \tag{2.6}$$

where $e(n) = 0$ for $n \leq 0$. Simulation runs are conducted to the P and PI controllers under the same lithography and circuit settings.The PID parameters are tabulated in Table 2.4.

Table 2.4: Control parameters for the three controllers

Controller	K_P	K_I	K_D
P controller	1.0×10^9		
PI controller	1.0×10^9	1.0×10^6	
PID controller	1.0×10^9	1.0×10^6	5.0×10^7

The step responses of these three controllers are plotted in Figure 2.7. The following paragraphs summarize the performance of these three controllers:

- P controller: Using the P controller, there exists a steady-state error.
- PI controller: With the implementation of integral control, the non-zero steady-state error is eliminated and the step response will converge to reference signal within 6 iterations.

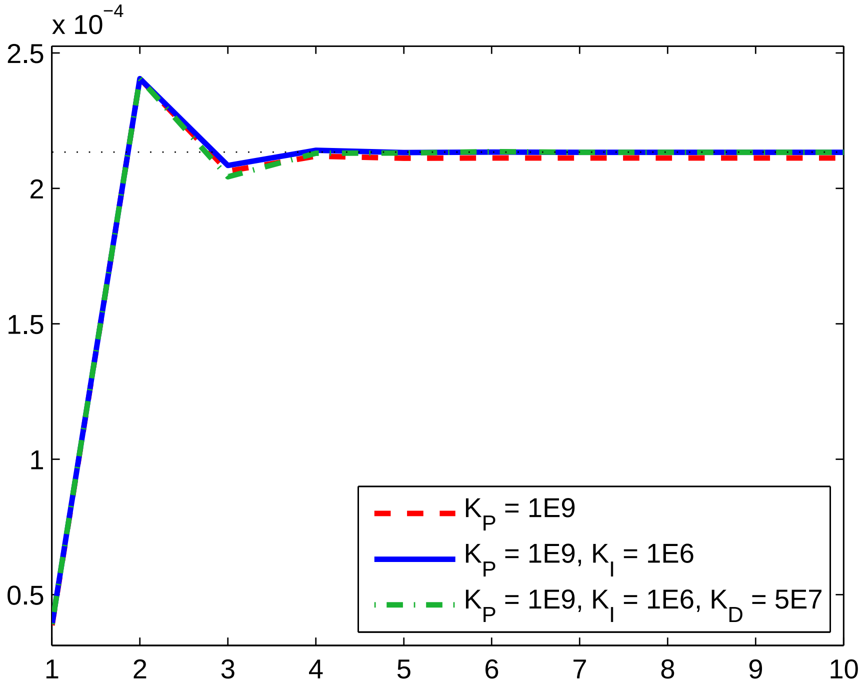


Figure 2.7: Step responses of P, PI and PID controllers

Further simulation runs are conducted to evaluate the impact of the integral term. Table 2.5 lists the PI parameters. The step responses are plotted in Figure 2.8. The mean square errors (MSE) of the first 30 cycles of these 6 PI controllers are calculated and tabulated in the last column of Table 2.5. From these simulation runs, it is observed that more overshoot will be introduced when large K_I is applied, *i.e.* $K_I \geq 1.0 \times 10^7$. This is because the integral term responds to accumulated errors from the past. On the other hand, if K_I is too small, *i.e.* $K_I = 1.0 \times 10^5$, non-zero steady-state error will not be fully eliminated.

- PID controller: In theory, derivative control can estimate the future errors and eliminate large overshoot. However, in this system, derivative

Table 2.5: Control parameters for the PI controllers

Controller	K_P	K_I	MSE ($\times 10^9$)
1	1.0×10^9	1.0×10^6	1.0379
2	1.0×10^9	1.0×10^8	1.0578
3	1.0×10^9	1.0×10^7	1.0393
4	1.0×10^9	1.0×10^5	1.0382
5	5.0×10^8	1.0×10^8	1.4255
6	5.0×10^8	1.0×10^7	1.2330

control does not have distinct advantage. Its step response is similar to PI controller. The derivative control term (D term) and the error signal of the first three iterations are tabulated in Table 2.6. The D term takes action from the 2nd iteration. Its symbol is the same as the error signal and the proportional control term (P term). Therefore in this system, D term only acts as an supplementary to proportional control.

Table 2.6: Error signal and derivative control term in PID control

n	$e(n)$	D term
1	1.74×10^{-4}	0
2	-2.72×10^{-3}	-1.01×10^8
3	9.08×10^{-2}	1.81×10^7

Further simulation runs are conducted for a wide range of K_D. Another 5 PID controllers are implemented. Table 2.7 lists the controller parameters for these PID controllers. The step responses are plotted in Figure 2.9. The mean square errors of the first 30 cycles of these 6 PID controllers are listed in the last column of Table 2.7. The system becomes unstable, when a large K_D is used, *i.e.* $K_D \geq 5.0 \times 10^8$. This is due to the over estimation of the future errors

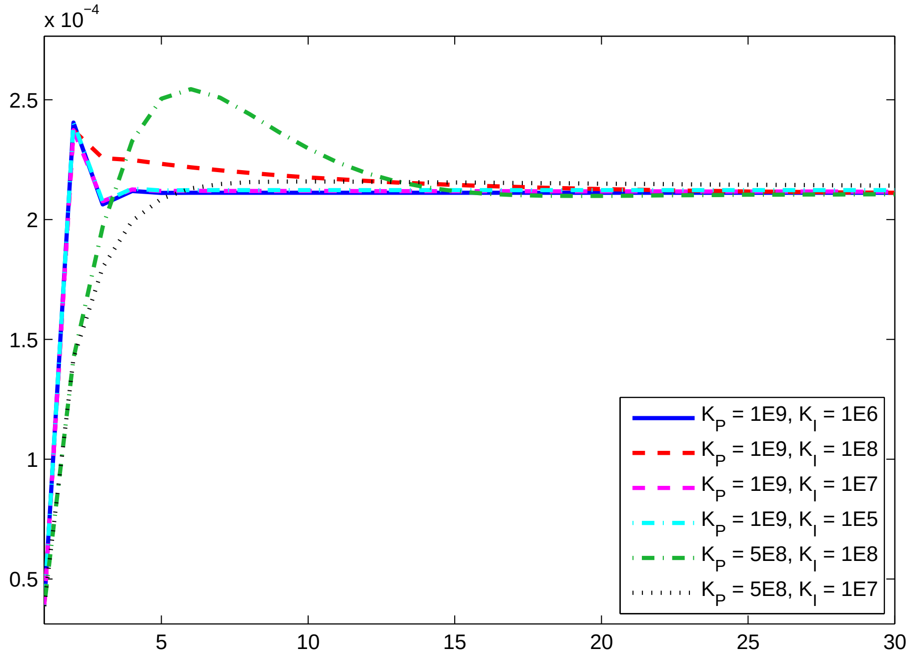


Figure 2.8: Step responses of the PI controllers

in derivative control. As stated in the previous paragraph, when small K_D is applied, *i.e.* $K_D \leq 1.0 \times 10^8$, there is no large difference in terms of step response from the PI controller.

Table 2.7: Control parameters for the PID controllers

Controller	K_P	K_I	K_D	MSE ($\times 10^9$)
1	1.0×10^9	1.0×10^6	0	1.0379
2	1.0×10^9	1.0×10^8	1.0×10^8	1.0588
3	1.0×10^9	1.0×10^8	5.0×10^8	6.6561
4	1.0×10^9	1.0×10^6	1.0×10^6	1.0387
5	5.0×10^8	1.0×10^6	1.0×10^8	1.0447
6	5.0×10^8	1.0×10^6	5.0×10^8	9.9622

The following conclusions can be drawn based on the aforementioned analysis. First, it is recommended to use PI only. Derivative control is not necessary in this application. Second, the IFT generated control parameters

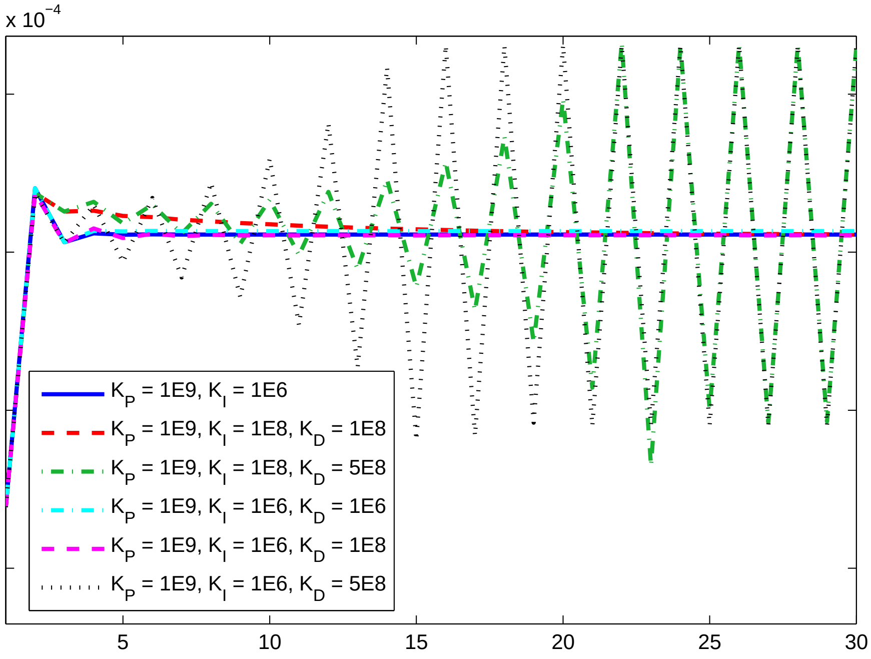


Figure 2.9: Step responses of the PID controllers

for the PI controller is optimal when compared to other manual settings with larger or smaller K_P and K_I.

2.6.3 Iterative Feedback Tuning Simulations

Simulations are conducted to evaluate the effectiveness of the IFT method when applied to TPO-OPC. The IFT simulations are conducted as follows:

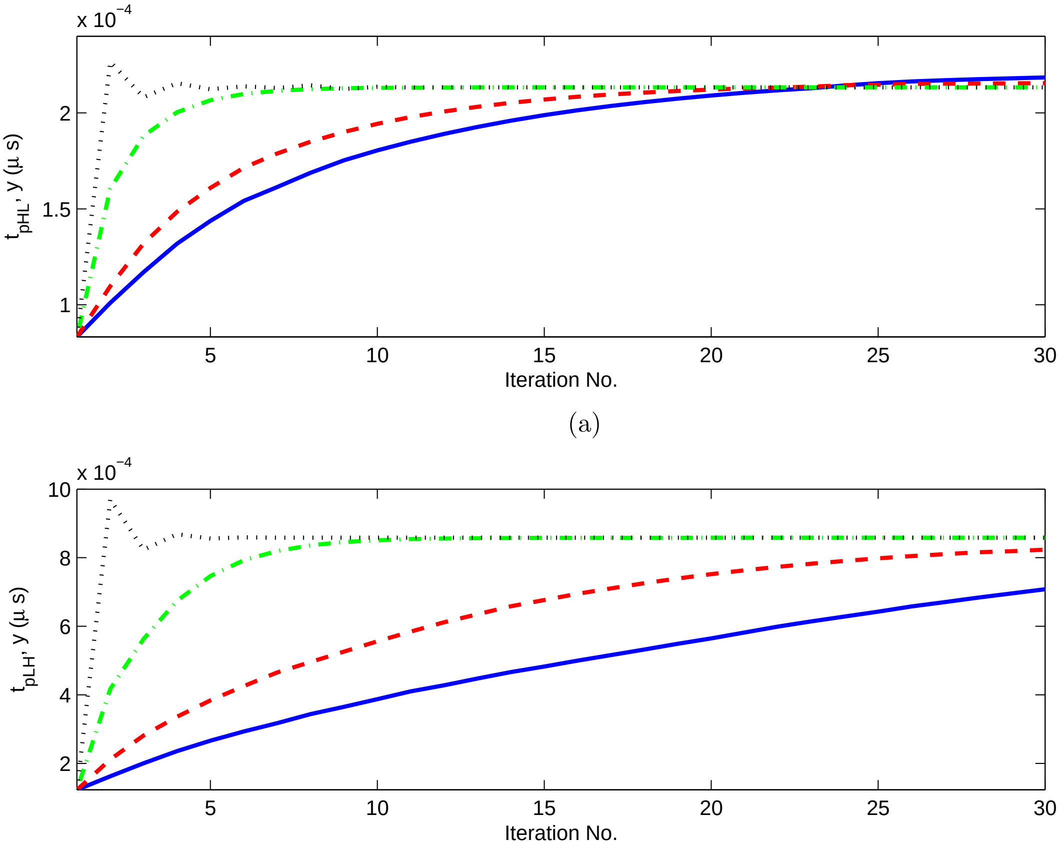
(1) Initialize controller parameters for all N sites (each transistor is treated as one site);

(2) Run IFT Run 1 and Run 2; [3]

(3) Update controller parameters for all N sites;

[3] Details are included in Appendix A.

(4) If $\dfrac{\partial J(\rho)}{\partial \rho}$ reduces to a desired number, optimum controller parameters are obtained; otherwise, repeat from step (2).



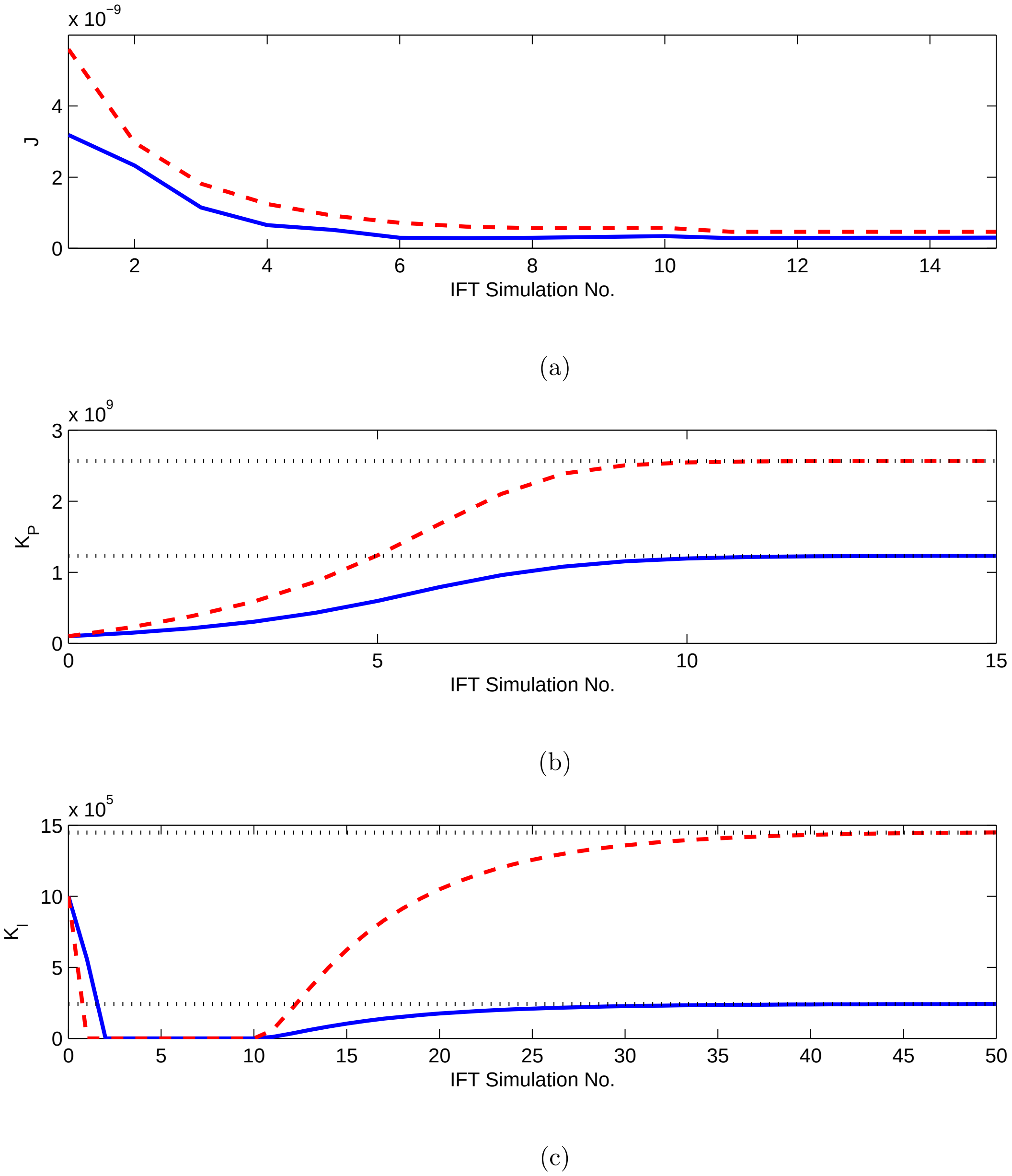
(a)

(b)

Figure 2.10: Comparison of step response with different sets of controller parameters for NMOS and PMOS sites respectively, solid line: before IFT, dashed line: after 1 IFT simulation, dash-dotted line: after 4 IFT simulations, dotted line: after 30 IFT simulations.

All controllers at N sites are initialized with $K_P = 1 \times 10^8$ and $K_I = 1 \times 10^6$. γ is set to 0.5 and η is fixed at 0. Since the whole mask design process is conducted in simulation, there is no penalty on the control signal magnitude and η can be set to 0. The first set of simulations is conducted on the standard cell NOR3. Figure 2.10 and 2.11 show the result of the IFT simulations. Since the performance with original controller parameters are unsatisfactory (solid lines in Figure 2.10(a) & 2.10(b)), one IFT simulation is conducted. The performance with updated controller parameters are plotted in dashed lines in Figure 2.10(a) & 2.10(b). The IFT simulations are repeated for multiple times and the performances after 4 and 30 IFT simulations are plotted in dash-dotted and dotted lines respectively. Better convergence can be observed. The cost functions of both sites after each IFT simulation are calculated and plotted in Figure 2.11(a). They converge to minimum within 10 cycles. Figure 2.11(b) and 2.11(c) show the convergence of controller parameters. K_P converges within 15 cycles and K_I takes up to 40 cycles to converge. In the simulations, K_I may be negative values due to large γ value in (A.4), so a truncation is set when K_I goes negative. Result shows that K_I will converge to non-negative numbers eventually.

Similar simulations are conducted on another 4 gates of which sizes range from 2 to 10 transistors (INV, NAND2, AND3, & OR4). For K_P, results show that there is little difference among different sites, and that NMOS and PMOS sites share K_P around 1.2×10^9 and 2.8×10^9 respectively. K_I of some gates converged to a positive number around 0.2×10^6 and 1.3×10^6 respectively for NMOS and PMOS sites, while some others converged to zero. This indicates that K_I are not necessary for some circumstances. However, to ensure zero steady state error, a positive K_I is preferred, *e.g.* $0.1 - 0.2 \times 10^6$ and $0.5 - 1.0 \times 10^6$ respectively for NMOS and PMOS sites. These results match with the choice of controller parameters in Section 2.6.1, especially for NMOS sites.



(a)

(b)

(c)

Figure 2.11: IFT simulations for NOR3. (a)-(c): convergence of cost function, controller parameters (K_P and K_I) respectively, NMOS site: solid line, PMOS site: dashed line.

Table 2.8 and 2.9 show the convergence time with controller parameters after a certain number of IFT simulations. Better convergence can be observed after a number of IFT simulations. It should be pointed out that IFT only needs to run once for each different optical and lithography setting to determine the optimum controller parameters K_P and K_I.

Table 2.8: Convergence time of NMOS sites (number of iterations to converge)

Gate	NMOS			
	Before IFT	After 1 IFT	After 4 IFT	After 30 IFT
INV	> 30	24	9	5
NAND2	> 30	25	11	6
NOR3	> 30	21	7	5
AND3	> 30	27	21	7
OR4	> 30	21	12	8

Table 2.9: Convergence time of PMOS sites (number of iterations to converge)

Gate	PMOS			
	Before IFT	After 1 IFT	After 4 IFT	After 30 IFT
INV	> 30	> 30	12	6
NAND2	> 30	> 30	16	5
NOR3	> 30	> 30	11	5
AND3	> 30	> 30	23	8
OR4	> 30	> 30	18	4

2.7 Conclusion

A timing performance oriented OPC method is presented in this chapter. Timing performance deviation from designed values has been reduced. Significant reduction (24 to 36%) in mask complexity as compared to conventional OPC methods are also achieved. In addition, a feedback controller is employed to improve the convergence speed. An iterative feedback tuning method is used to tune the PI controllers for improved convergence. The number of iterations is largely reduced compared to conventional OPC schemes without feedback controllers.

Chapter 3

Process Window Aware Optical Proximity Correction

3.1 Introduction

The previous chapter solved the electrical performance issue in OPC by using timing as the optimization index in the OPC feedback loop. However, only nominal conditions in lithography were considered and it may suffer from performance deviations under process variations. Process variations refers to the drift in process parameters during manufacturing. The iterative feedback tuning method proposed in the previous chapter only works in nominal lithography conditions and it cannot account for process variations. In this chapter, a process window aware OPC (PWA-OPC) method is designed and implemented. One key benefit of applying PWA-OPC is that, larger process window could be observed. In other words, this proposed method is more tolerable under process variations (a broader defocus and dosage variation tolerance) than other existing methods. A timing cost function is proposed to link the complex circuit behavior, *i.e.* propagation delay, with the target shape. Retargeting for the layout of each individual transistor is made possible by this cost function. Prior works in timing only focused on worst case delays of the standard cells [52, 53]. The proposed PWA-OPC attempts to optimize the accuracy of post-lithography timing performance of each delay case. In addition, a less aggressive OPC can be employed while accuracy of timing performance can be preserved. Therefore, mask complexity can be reduced significantly.

This chapter is organized as follows. Section 3.2 gives an overview on retargeting and performance based OPC methods. In Section 3.3, the PWA-OPC method is illustrated. Section 3.4 discusses simulation results, followed by conclusions in Section 3.5.

3.2 Overview of Retargeting and Performance Based OPC Methods

The literature consists of a number of model based retargeting methods [54–56]. In [54], a model based retargeting method was first presented to overcome the limitations of conventional rule based retargeting. Similar to the rule based OPC methods, rule based retargeting uses look-up tables in predefined retargeting libraries. Agarwal *et al.* [56] and Banerjee *et al.* [55] introduced methods to combine retargeting and OPC in one loop so as to optimize normalized image log slope (NILS). [1] Similar to the drawbacks of most conventional shape driven OPC approaches as stated in Section 1.3, all of the above-mentioned retargeting methods focused on the geometric aspects without taking into considerations the circuit performance.

An earlier work on device performance based OPC (PB-OPC) [22] demonstrated the feasibility and importance of matching circuit characteristics rather than shapes, and its impact on mask cost reduction. However, due to the complexity of circuit behavior, only the transistor drive current is optimized which does not fully account for the desired circuit behavior. For example, timing characteristics, such as propagation delay, are often a more desirable circuit behavior in digital logic gates [27]. Simulations reported a delay deviation of up to 4% for the simplest inverter [22]. Another drawback of this methods is that, process window is influenced due to less accurate timing performance. This chapter proposes a new timing cost function that relates timing performance of each gate and each case of the digital gates with mask patterns on the layout. This implementation overcomes the drawbacks of previous methods in terms of timing accuracy and process window.

[1] NILS is a metric in shape domain. NILS based methods are shape driven OPC, similar to EPE based methods. The NILS is a measure of the information content of the aerial image and represents an energy (intensity) gradient at the position of the nominal line edge [57]. Larger NILS means more information as to the proper position of the feature edge.

3.3 Methodology

3.3.1 Overall Flow

The block diagram of the proposed process window aware OPC (PWA-OPC) method is shown in Figure 3.1. The input to the system is the designed shape (layout) of a circuit and design timing. The design shape is generated by the IC designers. The design timing (design intent) is estimated from original hardware description language (HDL) or netlist using SPICE [47, 58]. The main body of the proposed PWA-OPC is an iterative loop. The loop iterates until the post-lithography timing converges to the original design timing. At each iteration, a timing driven retargeting (TDR) method is applied to conduct the retargeting process. A sparse OPC approach is then conducted on top of the target shape. The actual algorithms are implemented in a "PWA-OPC" macro block with is "TDR" plus "Sparse OPC" as shown in the main body of Figure 3.1. The shape after OPC is called the "mask layout". The mask layout of the final iteration is the output of the PWA-OPC flow. Intermediate mask layouts are subject to a lithography simulation. The output of the lithography simulation is called the "printed image", which is then subject to a "image2timing" process. [2] The difference between design timing and post-lithography timing, *i.e.* the error signal in Figure 3.1, is used as the input to the TDR in the next iteration.

[2] The "image2timing" refers to a post-lithography timing extraction using a method introduced by Ref. [36].

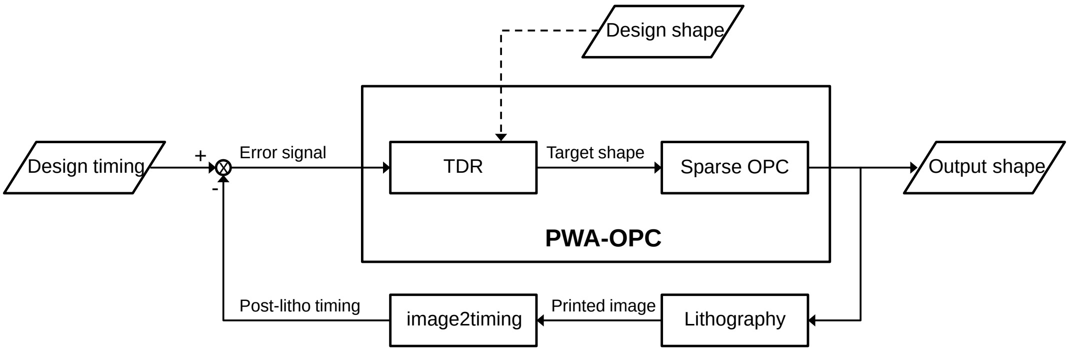


Figure 3.1: PWA-OPC block diagram

The implementation details of PWA-OPC are depicted in Figure 3.2. Step 4 to 9 is the detailed operations in TDR. The algorithm aims at minimizing the performance deviation error, `e(i)`, which is the difference between the post-lithography timing, $Post(\cdot)$, and design timing, $Design(\cdot)$. In this chapter, detailed "propagation delay" for each gate at all cases is used. [3] The threshold value, ϵ, determines the desired matching accuracy. The shift amount is a function, *i.e.* $\Gamma(\cdot)$, of `e(i)`, which can be a simple linear function or a complex feedback controller (details are given in Section 2.5).

3.3.2 Timing Driven Retargeting

3.3.2.1 Basic Idea

The layouts before TDR, within TDR and after TDR are shown in Figure 3.3. Figure 3.3(a) is the original design shape. Figure 3.3(b) shows the target

[3] Propagation delay is the length of time which starts when the input to a logic gate is valid, to the time that the output is valid. It is usually calculated using a transient analysis method. In the simulation, SPICE [47] is used as the calculator.

```
input  : design shape, designed timing
output: output shape
Load design shape and assign movable edges;
repeat
    updated ← false;
    foreach edge as edge(i) do
        e(i) ← $Post$(edge(i)) − $Design$(edge(i));
        if |e(i)| > $\epsilon$ then
            Compute the shift direction and amount: $\Delta W \& \Delta L$ ;
            Retarget edge(i);
            updated ← true;
    Apply sparse OPC;
    Conduct lithography simulation;
until updated = false ;
```

Figure 3.2: PWA-OPC algorithm

shape after TDR. TDR is actually to shift the edges of a transistor's target layout in vertical or horizontal directions. OPC is applied to the retargeted shape and the OPC mask layout is shown in Figure 3.3(c). The PWA-OPC method is compatible with state-of-the-art dense OPC engine, and a dense OPC can also be applied to the target shape from TDR block. [4] An example of a dense OPC mask layout is shown in Figure 3.3(d). The aim of TDR is to change effective width and/or length of the printed image of a transistor, and its electrical characteristics. [5]

When the transistor's width on the target shape is extended/shrunk by an amount of ΔW, the resulting post-lithography printed shape (after applying OPC and lithography) will change by an equivalent amount of $\Delta W'$, as indicated by [36], where $\Delta W' = f(\Delta W) = \Delta W + \delta$. δ is the

[4] Dense OPC can be applied to small devices with serious 2D distortions. For larger devices, sparse OPC is a better choice to reduce mask complexity.

[5] The transistor size (*esp.* cell height) is bounded within predefined areas, so that hazards such as touching power rails can be avoided and cell area will remain the same.

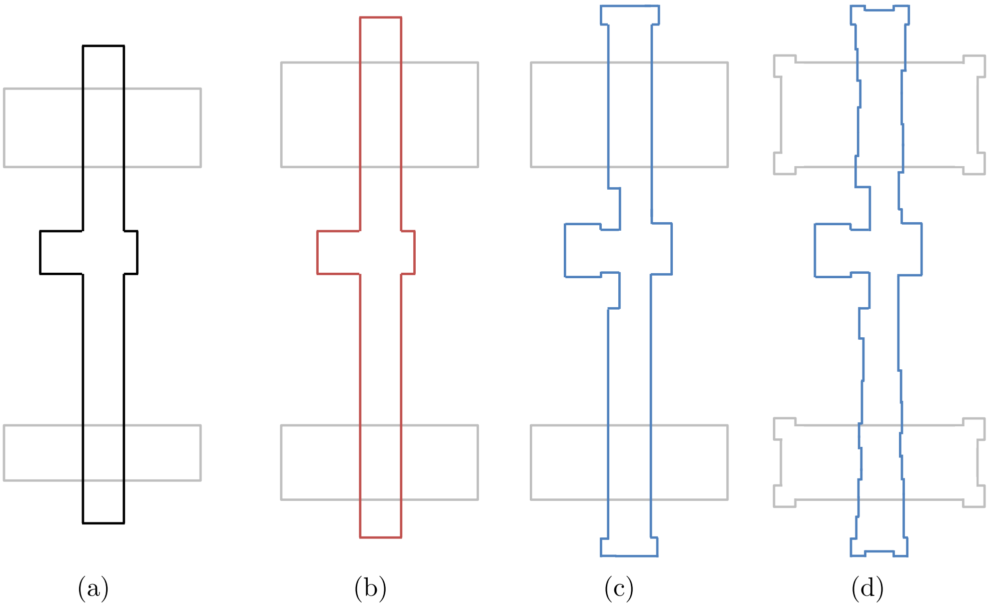


Figure 3.3: PWA-OPC input/output layout example: (a) original shape, (b) after retargeting, (c) after sparse OPC, (d) after dense OPC. Poly layers are colored and diffusion layers are gray-colored.

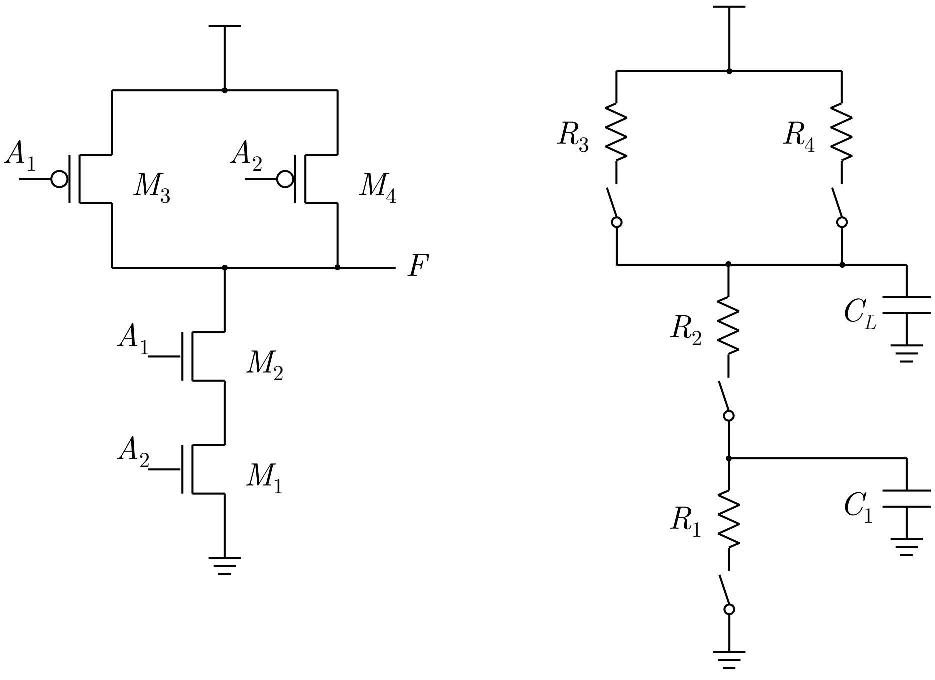
difference between $\Delta W'$ and ΔW caused by the lithography process. For an inverter, the propagation delay, t_p, will be affected according to (3.1):

$$\begin{cases} t_{pHL} = 0.52\dfrac{C_L}{(W/L)_n k'_n V_{DSATn}} \\ t_{pLH} = 0.52\dfrac{C_L}{(W/L)_p |k'_p V_{DSATp}|} \end{cases}, \tag{3.1}$$

where W is the transistor's equivalent width, L is the channel length, C_L is the load capacitance, V_{DSAT} is the saturation voltage, k' is a constant, and the subscripts, n & p, refer to NMOS and PMOS respectively. The aforementioned principle enables the tuning of propagation delay through target shape adjustment.

Similarly, when the transistor's length on the target shape is widen/narrowed by an amount of ΔL, the propagation delay is changed accordingly from (3.1).

3.3.2.2 Retargeting for Complex Gates



(a) (b)

Figure 3.4: Two-input NAND(NAND2): (a) schematic, (b) its RC-equivalent model

Employing propagation delay as design metric for the simple inverter in the proposed retargeting method is rather straightforward as explained earlier. However, the same approach cannot be applied to complicated gates. To understand the issue better, let's consider a two-input NAND gate shown in Figure 3.4. As it consists of 2 PMOS and 2 NMOS transistors, input data pattern dependency often results in different propagation delays, t_{pHL} and t_{pLH}. For the illustrated two-input NAND gate, there are 6 different possible propagation delays depending on the sizing of each transistors and the switching pattern as described by the following equations using Elmore's

[59] method:

$$
\begin{cases}
t_{pHL,00\rightarrow 11} = R_1 C_1 + (R_1 + R_2)C_L \\
t_{pHL,01\rightarrow 11} = (R_1 + R_2)C_L \\
t_{pHL,10\rightarrow 11} = R_1 C_1 + (R_1 + R_2)C_L \\
t_{pLH,11\rightarrow 00} = (R_3 \parallel R_4)C_L \\
t_{pLH,11\rightarrow 01} = R_3 C_L \\
t_{pLH,11\rightarrow 10} = R_3 C_L + (R_2 + R_4)C_1
\end{cases}
\tag{3.2}
$$

Any mask adjustment on any transistor will thus have different impact on delay accuracy as shown in Table 3.1. Therefore, no simple mask adjustment can be used to meet the design intent. This is the main obstacle which prevents the earlier PB-OPC method in [22] from adopting timing delay as the desired design metric. Table 3.1 also highlights the correlation of mask adjustment and delay accuracy. When target shape of M_4 is adjusted, it has significant impact on $t_{pLH,11\rightarrow 10}$, moderate impact on $t_{pLH,11\rightarrow 00}$, and almost negligible impact on other switching patterns. In fact, all the numbers highlighted in bold indicate significant correlation between target adjustment of the corresponding transistor and the delay deviation. It is the realization of this correlation which allows this work to propose a simple approach to overcome the above-mentioned obstacle as explained next.

Table 3.1: Change in propagation time (%) when target shape of cell NAND2 is extended by 10nm. Positive numbers means the delay is increased due to the fragment shift, while negative numbers means the delay is decreased.

Transistor	M_1	M_2	M_3	M_4
00 → 11	**-3.29%**	**-3.20%**	0.23%	0.20%
01 → 11	**-3.88%**	**-3.43%**	0.29%	0.03%
10 → 11	**-3.25%**	**-3.25%**	0.05%	0.21%
11 → 00	0.04%	0.30%	**-3.10%**	**-3.13%**
11 → 01	-0.06%	0.24%	**-1.16%**	0.02%
11 → 10	0.04%	0.30%	-0.04%	**-6.93%**

3.3.2.3 Adjusting Target Shape: Mapping Timing Arc to Target Shape

Inspired by (3.1), which indicates that delay is proportional to transistor length L and is inversely proportional to transistor width W, a new timing cost function based on a first order model is proposed to estimate the timing of complex gates:

$$D_i = D_{0,i} + \sum_{j \in transistors} \frac{\partial D_i}{\partial M_j} \Delta M_j + \sum_{j \in transistors} \frac{\partial D_i}{\partial L_j} \Delta L_j, \tag{3.3}$$

where $D_{0,i}$ is the original delay of the i_{th} timing arc [6] , D_i is the delay when transistor size changes (ΔM and ΔL) occur, $M_j = (W_j)^{-1}$ is the reciprocal of transistor width (since delay is inversely proportional to width), and L_j is the transistor length. SPICE [47] simulations are utilized to characterize the

[6] The "timing arc" here refers to the path from an input port to an output port with a certain input vector. Consider the example in Section 3.3.2.2, there are six timing arcs in a NAND2 gate.

sensitivity coefficients and result shows that the first order model is accurate within a reasonable range (±10%) of M and L.

It is found that ΔL results in significant change in timing delay and L tuning is thus employed for coarse delay tuning, whereas ΔM results in small delay change and it is employed for fine tuning. Hence, most of the time, the adjustment will be centered around the width while length adjustment only occurs occasionally. Therefore width and length adjustments are further extracted out of (4.5) to make the algorithm more efficient as follows:

$$\begin{cases} M_j = M_{0,j} + \sum_{i \in arcs} \frac{\partial M_j}{\partial D_i} \Delta D_i & \text{(fine tuning)} \\ L_j = L_{0,j} + \sum_{i \in arcs} \frac{\partial L_j}{\partial D_i} \Delta D_i & \text{(corase tuning)} \end{cases} \tag{3.4}$$

The above-mentioned relationship between timing arc and target shape (transistor width/length) can be exploited for target shape adjustment in the algorithm to determine the corresponding transistor and desirable target shape adjustment in vertical direction (ΔM) as follows:

$$\begin{aligned} \Delta M\left[\texttt{edge(i)}\right] = M_i - M_{0,i} &= \Gamma\left[\texttt{e}_M\texttt{(i)}\right] \\ &= \Gamma\left[\sum_{j \in arcs} \frac{\partial M_i}{\partial D_j}\left(t_{p,j,post-litho} - t_{p,j,design}\right)\right] \\ &= \Gamma\left(\sum_{j \in arcs} \frac{\partial M_i}{\partial D_j} t_{p,j,post-litho} - \sum_{j \in arcs} \frac{\partial M_i}{\partial D_j} t_{p,j,design}\right) \\ &= \Gamma\left\{Post_M\left[\texttt{edge(i)}\right] - Design_M\left[\texttt{edge(i)}\right]\right\}, \end{aligned} \tag{3.5}$$

where $\Gamma(\cdot)$ is the shift amount calculation function, and $\texttt{e}_M\texttt{(i)}$,

$Post_M\,[\texttt{edge(i)}]$ along with $Design_M\,[\texttt{edge(i)}]$ have been mentioned in Figure 3.2. [7] Since M does not directly have physical meaning, it is converted back to transistor width using:

$$W_i = M_i^{-1} = (M_{0,i} + \Delta M_i)^{-1} \tag{3.6}$$

Similarly, the shift amount in the horizontal direction (length adjustment) is:

$$\begin{aligned}\Delta L\,[\texttt{edge(i)}] &= \Gamma\,[\texttt{e}_L\texttt{(i)}] \\ &= \Gamma\left(\sum_{j\in arcs}\frac{\partial L_i}{\partial D_j}t_{p,j,post-litho} - \sum_{j\in arcs}\frac{\partial L_i}{\partial D_j}t_{p,j,design}\right) \\ &= \Gamma\left\{Post_L\,[\texttt{edge(i)}] - Design_L\,[\texttt{edge(i)}]\right\} \end{aligned} \tag{3.7}$$

However, it is only invoked when the width adjustment cannot cover the desired delay range.

3.3.2.4 Improvement from Previous Methods

As mentioned earlier, although the idea of this work is similar to Refs. [49, 50], there are two fundamental difference which improves the efficiency of the algorithm. Firstly, the two previous works adjusted L whereas this work adjusts W. From simulation studies, it is found that the timing delay is more sensitive to L adjustment than W adjustment. In the studies,

[7] In practice, to avoid disturbance by weakly correlated fragments, small $\frac{\partial D_i}{\partial M_j}$ factors can be omitted.

minimum step size of 1nm is adopted, which follows the manufacturing grid size requirement. For a 45nm inverter, an L adjustment will result in sensitivity of 3.3ps/nm. The minimum delay change of 3.3ps will correspond to 5.6% of inverter nominal delay of 58.4ps. On the other hand, a W adjustment will only result in sensitivity of 0.46ps/nm, which is about 7.1 times smaller than L adjustment. Secondly, to minimize the sensitivity due to L adjustment, Refs. [49, 50] have sliced the transistor region into multiple rectangle segments with individually adjustable L. In this case, at least 7 segments are required to achieve the same sensitivity as W adjustment. This implies that 7 variables need to be tuned in [49, 50] compared to 1 variable in this work in order to achieve similar delay sensitivity. Therefore, in addition to simple mask due to no fragmentation, this approach also results in more efficient tuning algorithm with fewer variables.

3.3.3 Sparse OPC

After TDR, the target shapes are subject to an OPC process. OPC is conducted based on a fragmented layout. The level of fragmentation can affect the performance of OPC. It is also related with mask cost. Low fragmentation (sparse OPC) is less expensive than high fragmentation (dense OPC) in terms of mask cost. On the other hand, the printed image quality of OPC with high fragmentation is usually better than OPC with low fragmentation. The following paragraphs will illustrate the effectiveness of difference levels of fragmentation by using indices such as process manufacturability index (PMI) and timing manufacturability index (TMI).

Ref. [60] proposed a process manufacturability index (PMI) to evaluate the quality of printed images. PMI of a set of layers can be expressed as:

$$PMI = \sum_{i \in layers} \frac{AREA\left(pvBand\left(i\right)\right)}{AREA\left(i\right)}, \tag{3.8}$$

where $AREA(\cdot)$ is the area of a layer or a region, and $pvBand(\cdot)$ is the process variability band under process variations. A simulation set on example circuits is conducted to measure PMI. The third column in Table 3.2 shows PMI of poly and diffusion layers of 4 masks, under same process variations. The 4 layouts are the printed images of a same set of target shapes after TDR. The only difference is the level of fragmentation in OPC. The average fragment size of the 4 OPC schemes are between 20-90nm (from dense OPC to sparse OPC). The table shows that PMI varies from 13% to 17%.

Similarly, a timing manufacturability index (TMI) can be defined. This TMI can be used to evaluate the timing performance of the printed image:

$$TMI = \frac{1}{n} \sum_{j \in arcs/paths} \frac{\left|t_{p,j,post-litho} - t_{p,j,design}\right|}{t_{p,j,design}}, \tag{3.9}$$

where n is the number of arcs or paths. The fourth column in Table 3.2 is the corresponding TMI of the 4 printed image layouts. As compared to the variation of PMI, the variation of TMI is much narrower. [8] Since geometric index such as PMI is not the ultimate goal, PMI can be sacrificed in exchange for savings in mask cost.

[8] Details of this relationship will be discussed in Chapter 4.

Table 3.2: PMI, TMI, and mask cost (normalized with respect to mask cost of layout 1) vs average fragment size

Layout	Avg. fragment size (nm)	PMI (%)	TMI (%)	Mask cost
1	21.2 (Dense)	13.24	0.3275	1.0000
2	43.6 (Less Dense)	14.01	0.3670	0.5036
3	65.3 (Less Sparse)	15.27	0.4024	0.3329
4	86.7 (Sparse)	17.26	0.4483	0.2536

The aforementioned example indicates that sparse OPC (with 80+nm per fragment) can achieve a same level of TMI as dense OPC (with 20+nm per fragment). About 4× mask cost reduction can be achieved when replacing dense OPC with sparse OPC. In Section 3.4, the performance of dense and sparse OPC will be compared.

3.3.4 Process Window Issue

Process window is defined as the range of process parameters within which predefined specifications can be satisfied. Conventional process window is usually related with geometry specifications such as critical dimensions (CD). This type of process window is often mentioned as "geometric process window" (GPW). In [61], an "electrical process window" (EPW) concept was proposed. In this work, EPW is updated with a "timing process window" (TPW). It refers to the range of process parameters within which no violation of timing occurs. [9]

A characterization method proposed in [62–64] can be employed to sketch the TPW. The post-lithography effective gate length (with normalized gate

[9] In practice, two process parameters are widely used: focus and dosage. Therefore, GPW, EPW and TPW are usually sketched on a 2D canvas such as Figure 3.6(a).

width) under process variation can be expressed as:

$$GateLength\,(f, d) = a_0 + a_1 f + a_2 d + a_3 f d + a_4 f^2 + a_5 f^2 d, \tag{3.10}$$

where f and d refer to focus and dosage values [64]. A TPW can be sketched based on the knowledge of the relationship between gate length and gate delay, which is introduced in Section 3.3.2.2. [10]

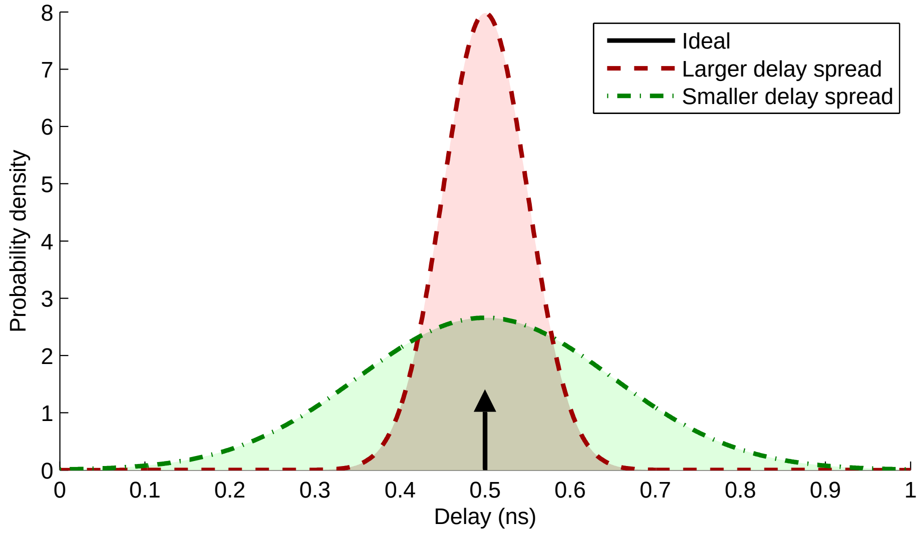


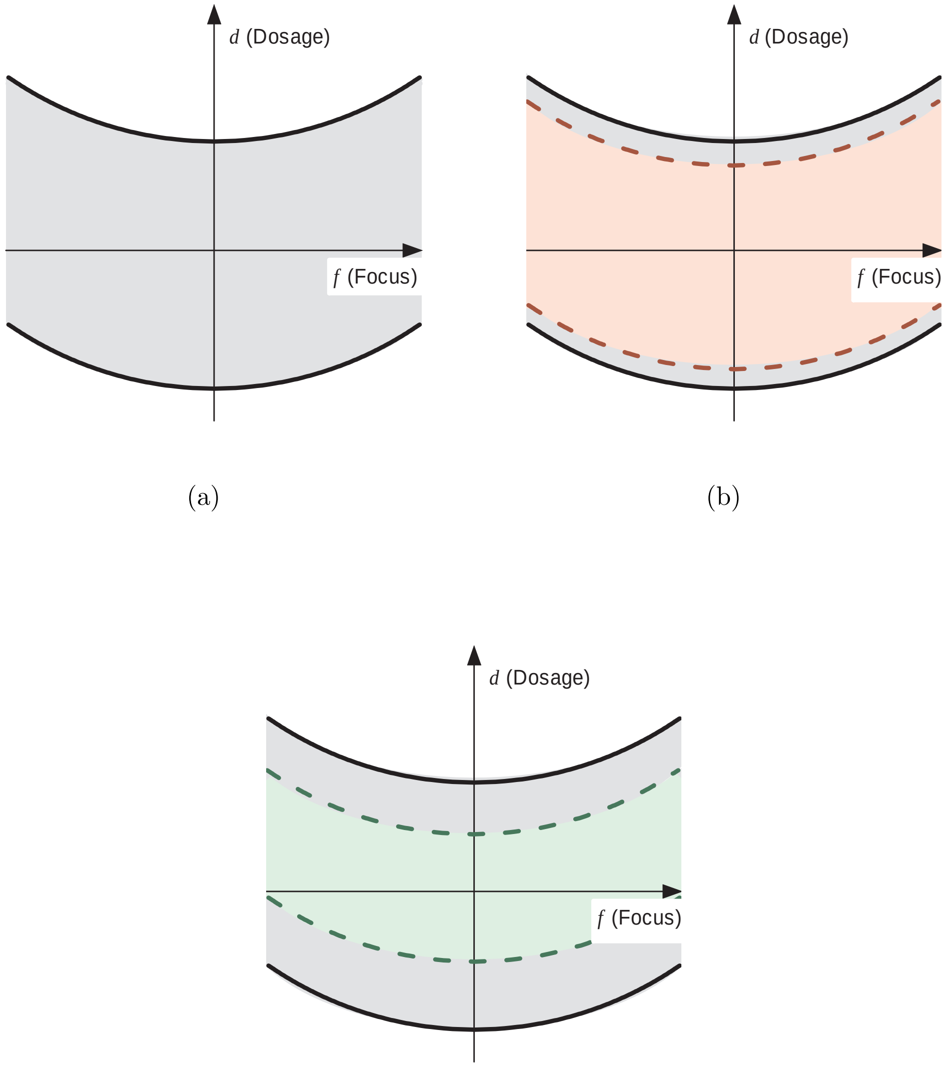
Figure 3.5: Three kinds of path delay probability distribution function

The following example shows the reason why PWA-OPC is more tolerable under process variations. [11] Three "Retargeting + OPC" schemes are considered in this example. The first is an "ideal" scheme which results in perfect timing performance, *i.e.*, all the gate delays are the same as designed values. The second and the third are two different non-ideal schemes. The gate delay deviations due to lithography at nominal conditions

[10] All the devices are assumed to sustain the same amount of gate length deviation due to process variations.

[11] Detail analytical proof can be found in Appendix B.

are depicted in Figure 3.5: the ideal case is a unit impulse, while the second has a narrower spread of gate delay than the third .



(a)

(b)

(c)

Figure 3.6: Timing process window example, (a) TPW of ideal case, (b) TPW of the scheme with smaller delay spread, (c) TPW of the scheme with larger delay spread.

For the ideal scheme, the TPW can be as large as the gray-colored area in Figure 3.3.4 (a-c). For the second and third schemes, however, the actual

TPW is smaller than the ideal TPW. Since gate delay spread of the second scheme is narrower, [12] the TPW of the second scheme (red-colored area in Figure 3.6(b)) will be greater than that of the third scheme (green-colored area in Figure 3.6(b)). This means the second scheme is more tolerable under process variations. The example indicates that narrower gate delay spread, which means more accurate timing, is a sufficient condition of a larger TPW. One of the main contributions of this work is that, the post-lithography timing of the proposed PWA-OPC method is much more accurate than other methods and the gate delay spread is narrower. This accurate timing results in a larger TPW.

3.4 Results and Discussions

3.4.1 Gate Level Simulation

The first simulation set is to evaluate the timing performance of the gates (standard cells). Simulations are conducted to compare among the proposed PWA-OPC approach, [22]'s method, an electrically driven method with considerations on timing performance [13], and a conventional shape driven method. In PWA-OPC, two different detailed OPC schemes are employed. The first OPC is a sparse rule based OPC and its average fragment size is 86.7nm. It is the preferred scheme in this work. The second OPC is a dense

[12] This work further assumes that spread of gate delay under small amount of process variations is dominated by retargeting and OPC itself rather than process variations.

[13] Idea of this method is from [52]. This method is employed and is modified to optimize each standard cell. An exhausting search is applied to find the optimum EPE-OPC parameters that provide best timing performance. This method is used in this work to fair compete with proposed PWA-OPC method.

model based OPC and its average fragment size is 21.2nm. It is used for comparison only. The 5 methods are referred as “PWA-OPC”, “PWA-OPC+Dense”, “PB-OPC”, “ED-OPC” and “SD-OPC” respectively. Design timing (design intent) is extracted beforehand. The last iteration post-lithography timing performances of the 5 methods are compared. The gate delay deviation of an arc j, e_j, is defined as:

$$e_j = \frac{t_{p,j,post-litho} - t_{p,j,design}}{t_{p,j,design}}. \tag{3.11}$$

The details of the e_j of all timing arcs in the whole standard cell library are shown in Table 3.3 and Figure 3.7. A distinct advantage of PWA-OPC method in terms of timing accuracy can be observed: the mean gate delay is smaller and the spread is also narrower. This shows the clear advantage in employing time delay as the direct metric in retargeting.

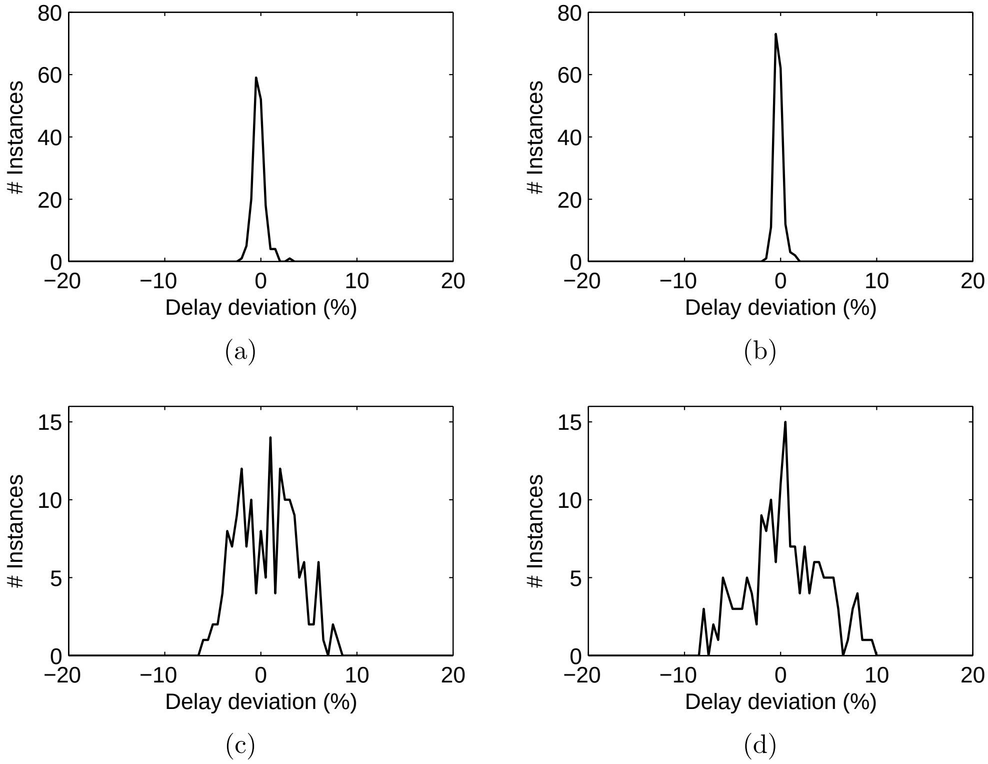
Table 3.3: Comparison of gate delay deviation

	Max $\lvert e_j\rvert$	Min $\lvert e_j\rvert$	Mean $\lvert e_j\rvert$ (TMI)	STD of e_j
PWA-OPC	3.19%	0.01%	0.45%	0.6245
PWA-OPC+Dense	1.86%	0.00%	0.33%	0.4451
PB-OPC	8.26%	0.02%	2.68%	3.0701
ED-OPC	9.94%	0.02%	3.12%	3.8965
SD-OPC	12.46%	1.24%	6.43%	5.6145

3.4.2 Circuit Level Simulation

The second simulation is conducted on larger circuits. [14] Four test circuits (c432, c499, c880 & c1908) from ISCAS’85 Benchmark [65] are chosen to be

[14] Larger circuits refer to practical digital application-specific integrated circuits (ASICs) with a number of logic gates. In this work, the circuit size ranges from 122 to 361 cells.



(a)

(b)

(c)

(d)

Figure 3.7: Histogram of gate delay deviation: (a) PWA-OPC; (b) PWA-OPC+Dense; (c) PB-OPC; (d) ED-OPC. Histogram of SD-OPC is even worse than (c) and (d) and is not shown in this series.

synthesized with 24 standard cells. They are also placed and routed using SoC Encounter [66] with the same configuration. The circuit level simulation evaluates “path delay”. Path delay refers to the delay from input port to output port (assume the full chip is a pure combinational logic). For each test circuit, a set of 2,000 random input patterns are excited into the input ports, and the rising or falling edge is captured on the specific output port to measure path delay. Results of 5 methods are compared. The 5 methods are referred as “PWA-OPC”, “PWA-OPC+Dense”, “PB-OPC”, “ED-OPC” and “SD-OPC” respectively.

The measured path delays are compared to design path delays. Figure 3.8

shows the histograms of path delay deviation of circuit c432 (histogram for other circuits are similar). PMI and TMI of all the 4 circuits are also listed in Table 3.4 and 3.5. Results in Figure 3.8 and Table 3.5 show that the path delay of the PWA-OPC methods outperform all other methods: the mean absolute path delay is smaller and the spread of path delay deviation is also narrower. [15] PMI of the PWA-OPC is also comparable to that of the other methods.

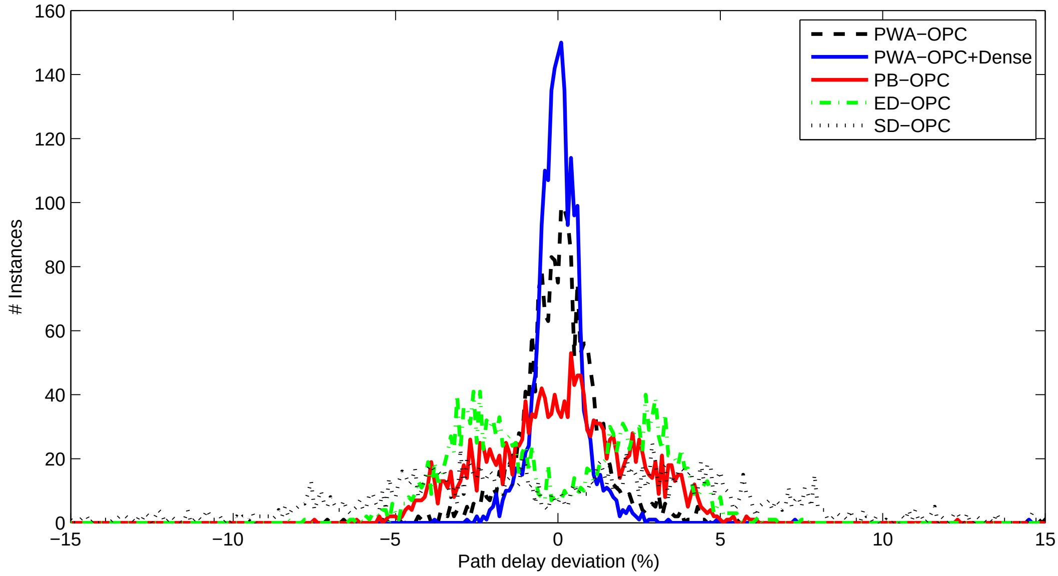


Figure 3.8: Histogram of path delay deviation, circuit c432

3.4.3 Sensitivity Test Under Process Variation

Figure 3.9 shows the gate delay deviation of the standard cells under variation of focus (defocus). Defocus conditions at -40nm and -20nm are compared with nominal focus. [16] Figure 3.10 shows the gate delay deviation of the standard

[15] Results of PB-OPC method in this work is better than those in [22] (mean: 3.47% for c1908). This is because a PI controller developed in Chapter 2 is used to improve convergence for PB-OPC method in this work.

[16] Results of +40nm and +20nm are similar to those of -40nm and -20nm, respectively.

Table 3.4: Comparison of PMI of full chip layouts (%)

Circuit	PWA-OPC	PWA-OPC+Dense	PB-OPC	ED-OPC	SD-OPC
c432	16.84	12.06	19.98	12.25	9.52
c499	16.03	12.20	21.50	14.30	9.89
c880	18.25	14.31	26.96	15.28	11.23
c1908	17.92	14.41	24.76	15.42	11.31
Average	17.26	13.24	23.30	14.31	10.49

Table 3.5: Comparison of TMI of full chip layouts (%)

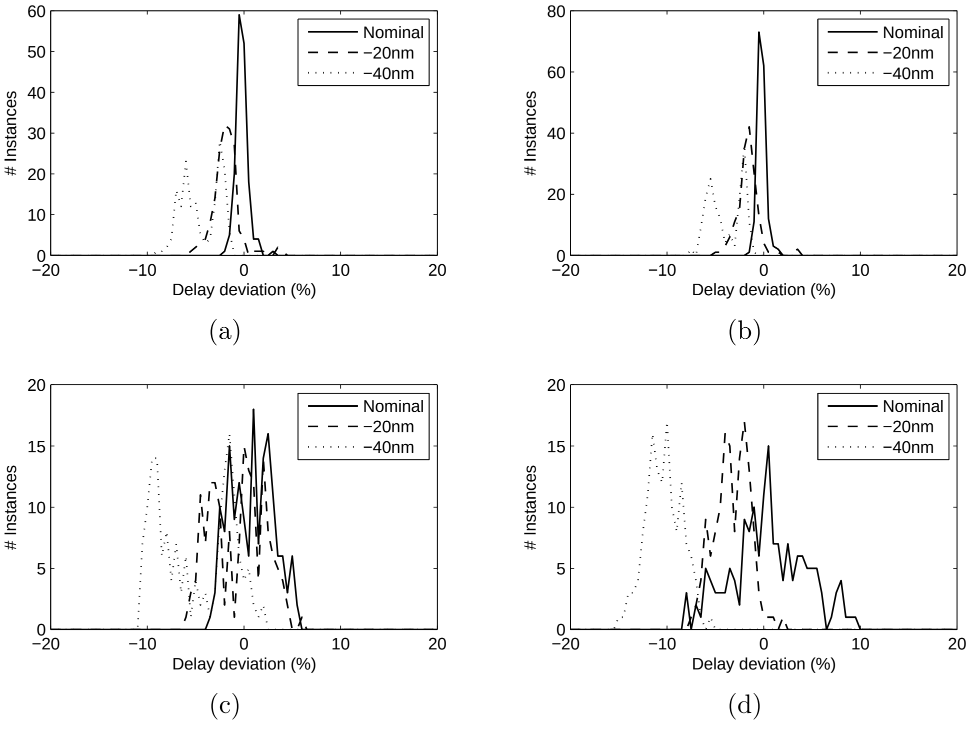
Circuit	PWA-OPC	PWA-OPC+Dense	PB-OPC	ED-OPC	SD-OPC
c432	0.82	0.53	1.74	2.54	4.72
c499	0.86	0.69	2.36	2.67	6.57
c880	1.17	1.00	2.06	2.85	4.88
c1908	1.02	0.78	2.17	3.10	6.21
Average	0.97	0.75	2.08	2.79	5.60

cells under three different dosage conditions, *i.e.* nominal, -3% and +3%. Result shows that the proposed PWA-OPC method outperforms PB-OPC and ED-OPC methods under process variations. PWA-OPC+Dense has a slightly better result than PWA-OPC in terms of spread of delay.

Table 3.6: Process window area (TPW area unit: $nm \cdot \%$)

	PWA-OPC	PWA-OPC+Dense	PB-OPC	ED-OPC	SD-OPC
$d_{min}(\%)$	-1.3	-1.5	-1.0	-0.9	-0.6
$d_{max}(\%)$	$+3.3$	$+3.6$	$+2.9$	$+2.6$	$+1.3$
$f_{min}(nm)$	-43	-48	-27	-25	-18
$f_{max}(nm)$	$+43$	$+48$	$+27$	$+25$	$+18$
TPW area	395.6	489.6	210.6	185.0	68.4

To better evaluate the timing process window (TPW), a series of



(a)

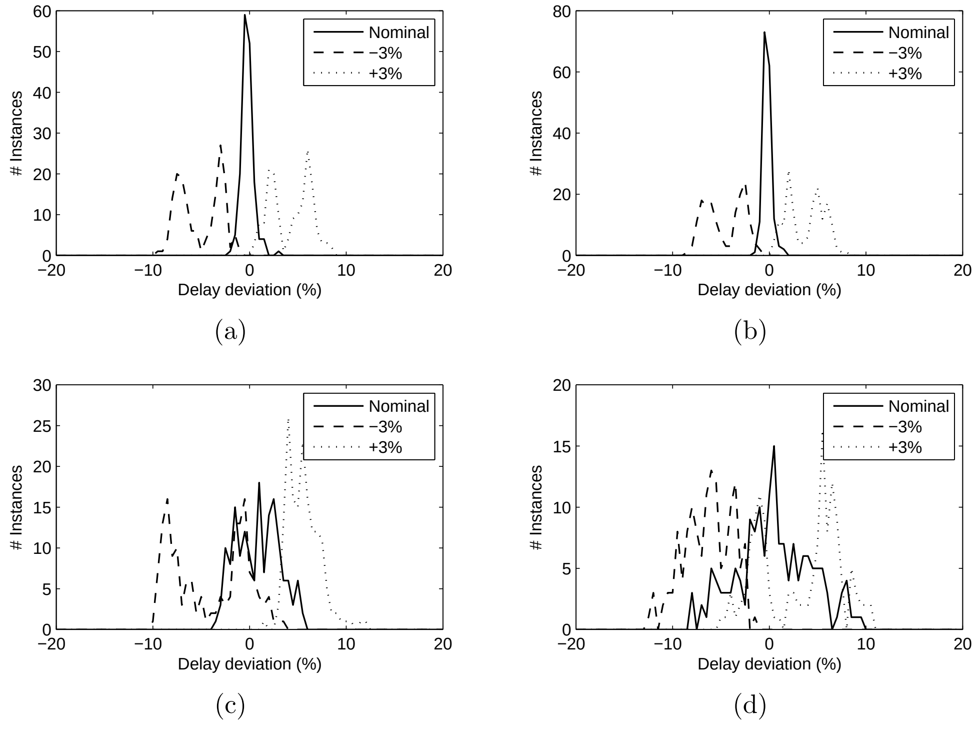
(b)

(c)

(d)

Figure 3.9: Histogram of gate delay deviation under focus variation: (a) PWA-OPC; (b) PWA-OPC+Dense; (c) PB-OPC; (d) ED-OPC.

simulations are conducted. For each of the 5 method compared in the previous subsection, an exhausting search with a step size of 1nm focus and 0.1% dosage values has been conducted to test the bound of focus and dosage values within which the timing specification is controlled, *i.e.* $D_L \leq D_{\pm 3\sigma} \leq D_U$, where $D_L = D_0 \times (1 - 10\%)$ and $D_U = D_0 \times (1 + 10\%)$. The area of timing process window, usually the largest inscribed rectangle within the lower and upper bounds as shown in Figure 3.11, can be measured from the above-mentioned search (multiple simulation sets) [11, 67]. Results of this simulation series are tabulated in Table 3.6. The TPW area of PWA-OPC+Dense is more than two times of PB-OPC. The TPW area of PWA-OPC is also 88% greater than PB-OPC. The TPW area



(a) (b) (c) (d)

Figure 3.10: Histogram of gate delay deviation under dosage variation: (a) PWA-OPC; (b) PWA-OPC+Dense; (c) PB-OPC; (d) ED-OPC.

of ED-OPC and SD-OPC is smaller than PB-OPC.

3.4.4 Mask Cost and CPU Run-time

To evaluate the mask size (or mask complexity), a method to count the number of fragments per transistor is employed. Fragment count implies mask complexity and is proportional to mask write time [23]. Although the effort in this work is mainly on poly layers, this simplification can also be extended to other layers. The poly fragment count with respect to non-OPC layout is tabulated in Table 5.1. PB-OPC has the least mask fragment count. However, its extremely simplified shapes are not practical beyond 65nm process. The mask size of PWA-OPC is remarkably smaller than PWA-OPC+Dense, ED-OPC, SD-OPC methods and other similar

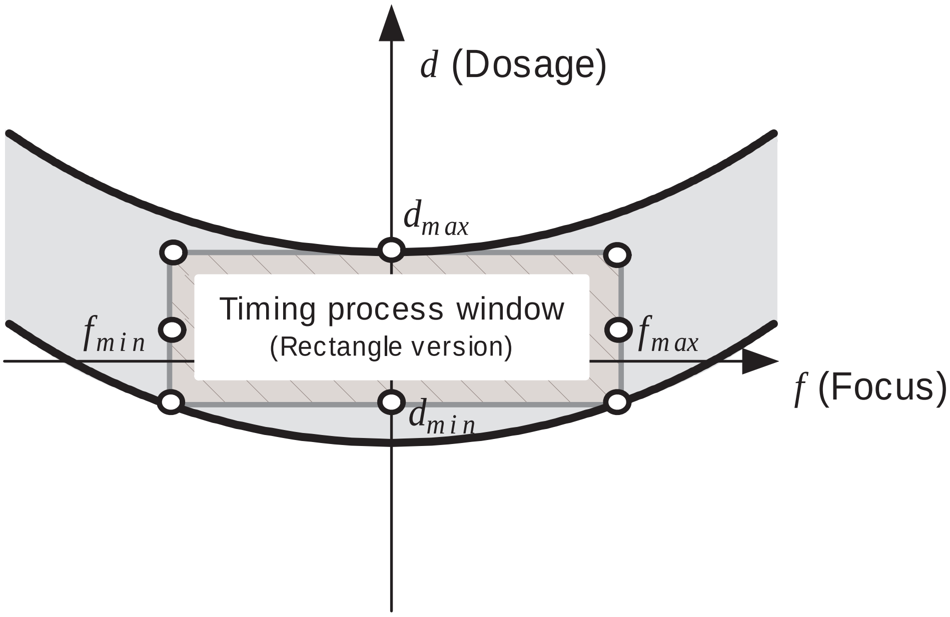


Figure 3.11: Illustration of rectangle version of timing process window: the largest inscribed rectangle ensuring no timing violation occurs at all the eight border points.

electrically driven OPC (ED-OPC) method reported in [21] (over 73% reduction).

Table 3.7: OPC mask fragment count (normalized with respect to Non-OPC)

Scheme	Normalized mask fragment count
Non-OPC	1.0
PWA-OPC	5.3
PWA-OPC+Dense	19.8
PB-OPC	2.2
ED-OPC	22.6
SD-OPC	21.3
ED-OPC in [21]	22.4

Table 3.8: CPU run-time (normalized with respect to SD-OPC [a])

Scheme	Normalized CPU run-time	Increment
PWA-OPC	0.87	-13%
PWA-OPC+Dense	1.66	+66%
PB-OPC	0.80	-20%
ED-OPC	1.45	+45%
SD-OPC	1.00	0%

[a] CPU run-time of SD-OPC is 185s.

The overall CPU run-time of the 5 methods on circuit c432 are tabulated in Table 3.8. The overall run-time of PWA-OPC is shorter than ED-OPC and SD-OPC. This is because the timing goal of PWA-OPC requires fewer iterations than others' geometric goal. PWA-OPC+Dense consumes more CPU run-time due to its extensive OPC computation. PB-OPC is even faster because it's also based on electrical goal.

3.5 Conclusion

A process window aware OPC technique using timing delay as the design metrics is proposed and implemented. Due to the complexity of timing behavior of complex gates, this is only possible through the newly proposed cost function, which successfully correlates the printed mask shape with the time delay parameter. In addition, the accurate timing performance allows larger tolerance under process variations and a twice larger process window from previous methods can be observed. Due to the sparse OPC approach, significant mask cost reduction up to 73% has also been achieved.

Chapter 4

Optical Proximity Corrected Mask Simplification Using Over-designed Timing Slack

4.1 Introduction

The timing performance issue of OPC has been addressed earlier by integrating timing metric in the feedback loop in OPC. This applies well when the manufacturing side permits modification to the OPC plant. However, in some cases, manufacturers are only allowed to do minor changes after a compulsory conventional OPC [18]. One solution is to do substraction or simplification after the OPC is completed. In this chapter, a methodology is designed and implemented to simplify mask patterns after OPC by using the extra margin in timing performances (over-designed timing slack). This methodology can be applied after a conventional OPC run, and is compatible with the current application-specific integrated circuit (ASIC) design flow.

To better understand the feasibility of the mask simplification method, an approach to characterize the relationship between timing yield and OPC mask cost is proposed. Through a mask utility function, optimal OPC schemes can be chosen to achieve a lower mask cost and a better timing yield. This motivates the idea to conduct mask simplification which incurs little penalty on timing yield.

The mask simplification method is applied to each occurrence of over-designed timing slack. The timing cost function proposed in Chapter 3 is utilized in this work to map timing slack in timing domain to mask patterns in shape domain. This enables mask patterns to be adjusted selectively based on the outcome of the cost function. When compared to existing OPC methods without mask simplification in the literature, this approach achieved a 51% reduction in mask fragment count and 13-20% reduction in MEBES file size, which leads to a large saving in lithography manufacturing cost. The result also shows that timing closure is ensured, though part of the timing slack has been sacrificed. [1]

This chapter is organized as follows. Section 4.2 gives an overview on timing yield and over-designed timing slack. A characterization method is demonstrated in Section 4.3. In Section 4.4, the OPC mask simplification

[1] Timing closure refers to the status that a design meets its timing requirements.

method is illustrated. Section 4.5 discusses simulation results, followed by conclusions in Section 4.6.

4.2 Timing Yield and Over-designed Timing Slack

The cost of OPC mask is usually related to the complexity of the mask polygons, which is also known as the aggressiveness of an OPC scheme [23]. Dense OPC schemes with high cost are expected to achieve good printed image quality on silicon, while sparse OPC schemes with lower cost often result in poor printed image quality on silicon. However, good printed image quality is not the ultimate goal of the semiconductor manufacturing process. In fact, timing yield is a more desirable metric for microchip [26]. Therefore, it is important to evaluate whether gain in timing yield is worthy of an aggressive OPC scheme to avoid unnecessary mask cost. The literature consists of a few works on the issue of statistical timing analysis on lithographic simulation [53, 68]. However, Refs. [53, 68] only evaluate the timing yield itself, while little effort has been paid to explore timing yield with respect to different levels of OPC aggressiveness. Research gap exists in the issue of the relationship between mask cost and timing yield. Therefore, it is of great interest to explore such relationship.

In practical integrated circuits, over-designed timing slack exists. It can contribute to reduction in mask cost, if a link between mask cost and timing yield is found. In the state-of-the-art ASIC design flow, a circuit is typically built on standard cell libraries [27]. In digital circuit design, standard cells are usually designed with excessive timing slack taking into consideration of various process, voltage and temperature (PVT) variations during run-time. This extra margin in timing slack has not been well utilized in the past. Although over-designed timing slack was studied and reported in [69, 70], the potential to exploit it for manufacturing cost reduction has not been investigated. On the other hand, OPC is commonly adopted to produce desired printed image quality on silicon rather than desired timing delay of a certain digital cell. This often led to increase in mask complexity and cost. Since mask complexity accounts for about a third of the total manufacturing cost, it is of interest to explore the potential of mask simplification by exploiting the excessive timing slack of a conservative digital cell [52, 68, 70].

The literature consists of several works on the over-designed timing slack [49, 52, 68–70]. Pioneer works in [52, 69, 70] proposed methods to measure or estimate timing yield. A circuit based method was invented to reduce the process variation induced timing uncertainty [68]. In [49], a method to convert timing slack to shape slack was proposed to enhance process window. None of the previous works took mask cost into consideration, and the over-designed timing slack was discovered but not utilized. Ref. [14, 24] tried to reduce mask cost with consideration of circuit performance, but these two are all based on nominal lithography conditions without process variations.

4.3 Characterizing of Timing Yield and Manufacturing Cost for Optical Proximity Correction Masks

In this section, a method to characterize the relationship between timing yield and OPC mask cost is proposed. Through a mask utility function, optimal OPC schemes can be chosen to obtain lower mask cost and good timing yield. This motivates the later parts of this chapter to conduct OPC mask simplification.

4.3.1 Problem Formulation

Timing performance of a circuit is impacted due to the variations in process, voltage and temperature (PVT). Timing yield at a clock period T can be defined in terms of a cumulative distribution function, P: [2]

$$\text{Yield}(T) = P\left(\text{Slack}(T) > 0\right) = P\left(\min_i \left(T - t_{p,i} - t_s\right) > 0\right), \tag{4.1}$$

where $t_{p,i}$ is the path delay of path i under PVT variations, and t_s is the setup time of the registers. Process variations can be modeled as:

$$X_{total} = X_{inter-die} + X_{intra-die}, \tag{4.2}$$

where $X_{inter-die}$ and $X_{intra-die}$ denote die-to-die and within-die variations, respectively. The die-to-die variations are assumed to have random variations. The within-die variations can be decomposed into systematic and random

[2] Clock skew is ignored in this work, since its impact on lithography induced variation in timing yield is insignificant.

variations. The systematic variation is caused by the layout dependency of the process and OPC induced effects. The random variation is due to process fluctuations around its nominal value.

Table 4.1 shows an example of timing yield and mask cost of a same circuit with three different OPC recipes. "Perfect OPC" means the post-lithography image (image in silicon) is identical to the designed layout, and the data of "Perfect OPC" is used only for reference. The remaining two (OPC 1-2) are real non-perfect OPC recipes with same variations of PVT parameters, and the only difference is the aggressiveness of OPC. It is clear that OPC 2 is worse than OPC 1 in terms of timing yield. However the benefit of applying OPC 2 is that its mask cost is only 40% of OPC 1. The purpose of this work is to propose a mask utility function to rate the quality of an OPC mask so that both timing yield and mask cost are considered, and details will be introduced in the following sections.

Table 4.1: Timing yield and mask cost of three OPC recipes, $T_0 = 46.5ns$

OPC recipe	Yield($0.8T_0$)	Yield($0.9T_0$)	Yield(T_0)	MaskCost
Perfect OPC	90.0%	96.6%	98.9%	-
OPC 1	84.5%	92.2%	96.5%	1.0
OPC 2	75.5%	86.5%	93.0%	0.4

4.3.2 Characterization Method

The overall flowchart of the proposed framework is depicted in Fig. 4.6. An OPC mask is generated following standard digital design flow and OPC recipe using commercial software. The resulting OPC mask is then subjected to timing yield estimation and mask cost estimation. Based on these estimations, mask utility function is performed with its result fed back to the framework to adjust the OPC recipe. The output (or optimal) mask is the OPC mask with maximum value of mask utility function, with a predefined weighting factor. Mask utility function of mask i at a working clock period T is defined as:

$$U_i = k_y \mathrm{Yield}_i(T) - k_m \mathrm{MaskCost}_i, \tag{4.3}$$

where k_y and k_m are two positive constants, and $\mathrm{Yield}_i(T)$ can be calculated using Eq(4.1). It is actually conducted using Monte-Carlo simulations with the information on design spec and netlist. For a given OPC mask, we can first simulate its post-lithography image under lithographic process variations. Its post-lithography timing, *i.e.* $t_{p,i}$ of all paths, can be simulated using the method in Ref. [22]. Mask cost can be estimated using the method in Ref. [23]:

$$\mathrm{MaskCost}_i = w_0 + w_1 V_i + w_2 L_i, \tag{4.4}$$

where w_0, w_1, w_2 are obtained using linear regression with real data from the fab, and V_i is vertex count of mask polygons and L_i is total line edge (TLE). Therefore the mask utility function can be calculated using Eq(4.3).

4.3.3 Simulation Results

To validate our proposed framework, simulations on Nangate 45nm Open Cell Library are conducted. Mentor Graphics Calibre is employed as OPC mask generator and lithography simulator. A 193nm light source with 1.33 hyper-NA immersion lithography is used. Five test circuits are selected from

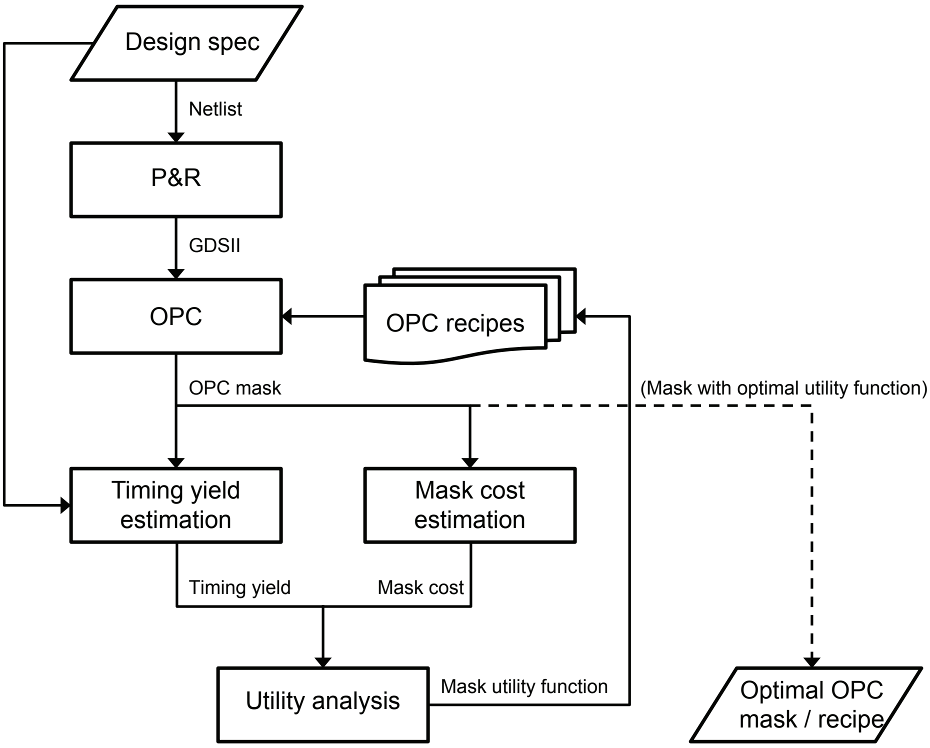


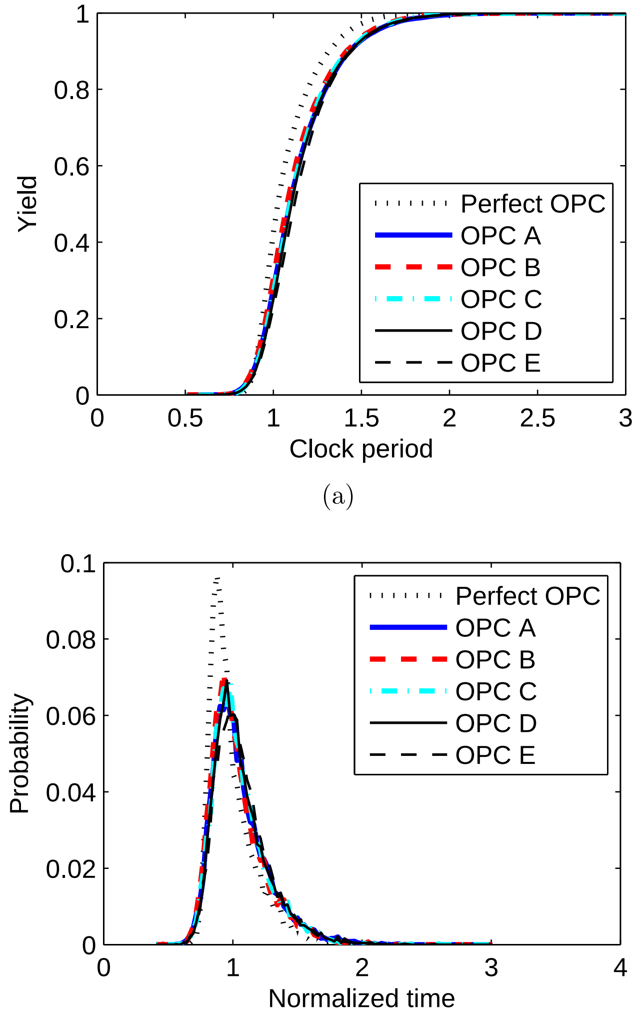
Figure 4.1: Flowchart of proposed characterization method

ISCAS'85 benchmark. A set of five OPC recipes are employed, and their average fragment sizes are tabulated in Table 4.2. These five recipes are with similar geometry constraints and process window information.

Table 4.2: Average fragment size of five OPC recipes

OPC recipe	OPC A	OPC B	OPC C	OPC D	OPC E
Avg. fragment size (nm)	36.07	45.26	65.58	80.90	90.59

The path delay PDF plot of circuit c432 is shown in Fig. 4.2(a). The corresponding timing yield is shown in Fig. 4.2(b). The five non-perfect OPC recipes (OPC A-E) are arranged in ascending aggressiveness. OPC A is the most aggressive and OPC E is the least aggressive. It can be seen that the path delay spread of non-perfect OPC recipes is wider than "Perfect OPC". The yield of non-perfect OPC recipes is also worse than "Perfect OPC" due to the optical proximity effects.



(a)

(b)

Figure 4.2: Path delay PDF and timing yield of OPC A-E (c432)

The detail yield and mask cost calculated using Eq(4.1) and (4.4) are shown in Fig. 4.3. Mask cost of the five non-prefect OPC recipes are normalized with respect to original designed layout. The variation in mask cost (up to 60%) when OPC recipe changes is much larger than the variation in timing yield (around 4%). This indicates that it is expensive to gain high timing yield by using aggressive OPC recipes.

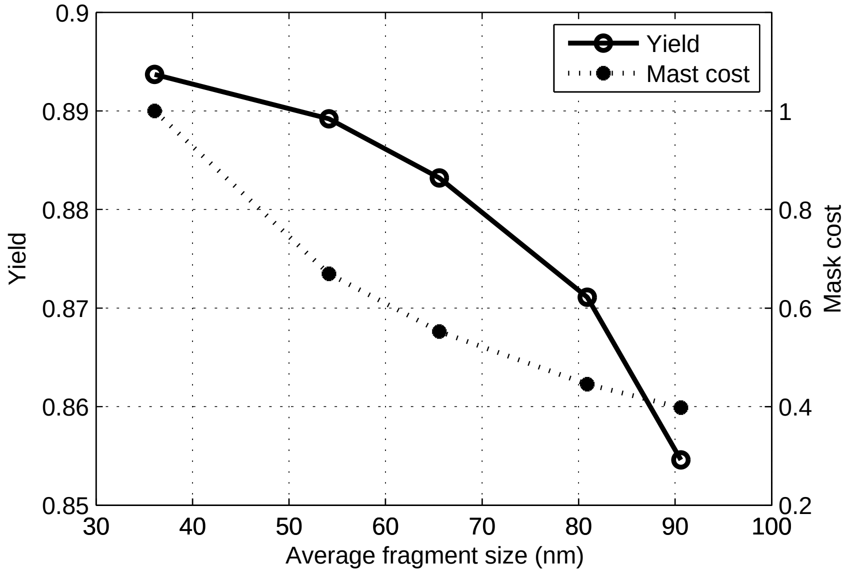


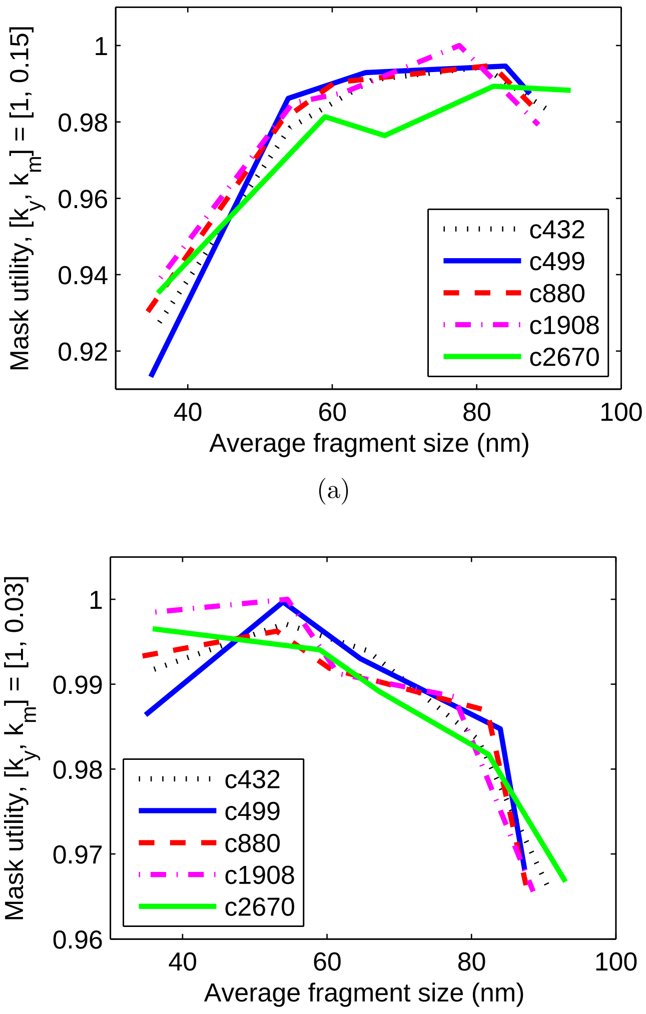
Figure 4.3: Timing yield and mask cost (c432)

Mask utility of c432 is calculated to evaluate the performance of a mask, using Eq(4.3). The values are tabulated in Table 4.3. When calculating mask utility, we adopted two sets of parameters: (a) $[k_y, k_m] = [1, 0.15]$, and (b) $[k_y, k_m] = [1, 0.03]$. These two sets place emphasis on mask cost and timing yield respectively. OPC D has the largest utility when mask cost is more desirable, while OPC B has the largest utility when timing yield is more desirable.

Table 4.3: Normalized mask utility of c432

OPC recipe	OPC A	OPC B	OPC C	OPC D	OPC E
Mask utility ($U_{i,a}$)	0.9275	0.9784	0.9910	0.9944	0.9822
Mask utility ($U_{i,b}$)	0.9918	0.9971	0.9940	0.9835	0.9661

To better validate our approach on a wider range of circuits, similar simulations are conducted on another four ISCAS'85 benchmark circuits, and the mask utility is evaluated for each circuit respectively. Result in Fig.



(a)

(b)

Figure 4.4: Normalized mask utility of 5 circuits

4.4 shows that the mask utilities of all the five circuits have similar trend. When we choose $[k_y, k_m] = [1, 0.15]$ (cheaper mask cost preferred), the mask utility points out OPC C-E, which are less aggressive. On the contrary, if we choose $[k_y, k_m] = [1, 0.03]$ (higher timing yield preferred), OPC A and OPC B become more viable.

In practice, mask cost occupies about 20% of the overall semiconductor manufacturing cost [3]. Hence, a mask cost reduction of 60% will directly translate to manufacturing cost saving of 12%. On the other hand, as the manufacturing cost is inversely proportional to yield, the overall cost increment is about 5% when the yield decrease from 89% to 85% (the example in Fig. 4.3). This illustrates the potential gain of trading of yield with mask complexity. For the proposed algorithm, we can choose the optimal $[k_y, k_m]$ setting to result in minimum manufacturing mask cost as follows: $k_m = k_y \hat{C_m} \hat{C_o}^{-1}$, where $\hat{C_m}$ is the estimated total mask cost and $\hat{C_o}$ is the estimated total manufacturing cost. Therefore, we can use $[k_y, k_m] = [1, 0.20]$. The result will be similar to Fig. 4.2(a), which means that less aggressive OPC schemes (OPC C-E) are more preferable.

4.4 OPC Mask Simplification Methodology

Simulation result in the previous section motivates this work to implement a mask simplification method using over-designed timing slack. This mask simplification method can reduce mask cost significantly with little sacrifice in timing performance. As shown in the probability density function (PDF) plots in Figure 4.7, timing slack usually occurs at both best and worst cases of a circuit's timing performance. Simulations show that the amount of total timing slack can be up to 30% of the nominal delay. Hence, part of this over-designed timing slack can be traded-off to reduce the mask cost. It has been reported that the aggressiveness of OPC schemes can impact the printed shapes and timing performance. A less aggressive OPC scheme may result in less perfect printed image and reduced timing slack (e.g. "PDF-2" in Figure

4.7). If the aggressiveness of OPC scheme is adjusted in such a way that the worsen timing performance only reduces the available timing slack without impacting the timing closure of the circuit, the outcome will be a simplified mask with reduced cost.

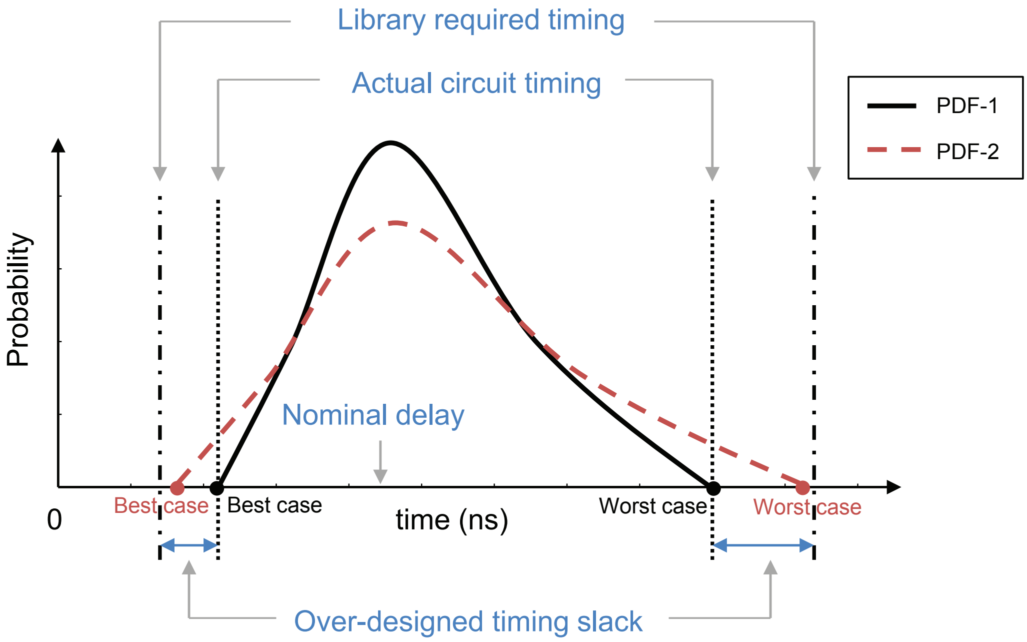


Figure 4.5: Over-designed timing slack found in the path delay probability density function (PDF) plot. PDF-1: complex mask with better timing; PDF-2: less complex mask with reduced timing slack but still meets library requirement.

The proposed mask simplification flow is shown in Figure 4.6. Implementation details of this method are illustrated in Figure 4.8. The inputs include the original OPC mask from conventional OPC process and the library timing of each timing arc pre-characterized using SPICE simulations. The "timing arcs" of a standard cell here refer to the paths from input ports to output ports. During each iteration, the timing slack, which accounts for the difference between timing arc estimated based on

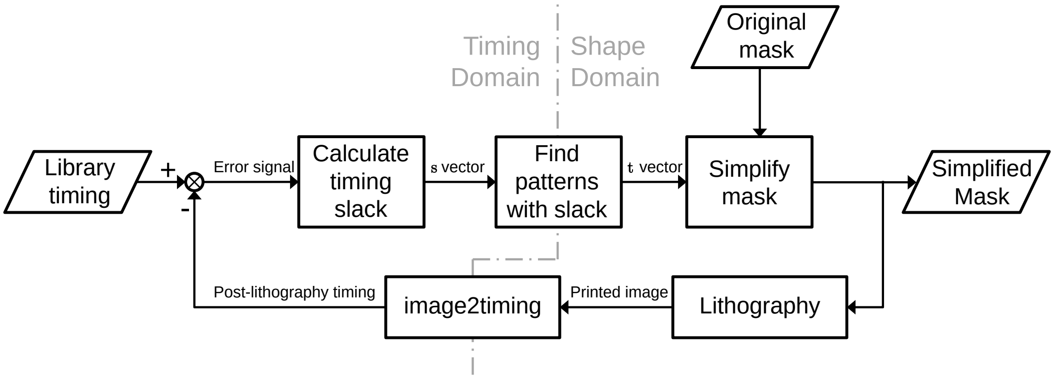


Figure 4.6: Flowchart of proposed OPC mask simplification method

digital cell library timing and timing arc estimated based on post-lithography timing, are calculated. For post-lithography timing estimation, printed silicon image through lithography simulation combined with non-rectangular transistor slicing technique are used. For those timing arcs with positive timing slack, the digital cell involved in the timing arc estimation will be subjected to mask simplification process (fragment merge) based on a timing cost function. The relationship between digital cells and timing arcs is found using the timing cost function proposed in this work. It is illustrated in Figure 4.7. The mask simplification procedure is repeated until no further reduction in timing slack is possible.

4.4.1 Timing Cost Function

In Chapter 3, a first order model was proposed to estimate the delay of complex CMOS gates:

$$D_j = D_{0,j} + \sum_{k \in Trans} \frac{\partial D_j}{\partial M_k} \Delta M_k + \sum_{k \in Trans} \frac{\partial D_j}{\partial L_k} \Delta L_k, \tag{4.5}$$

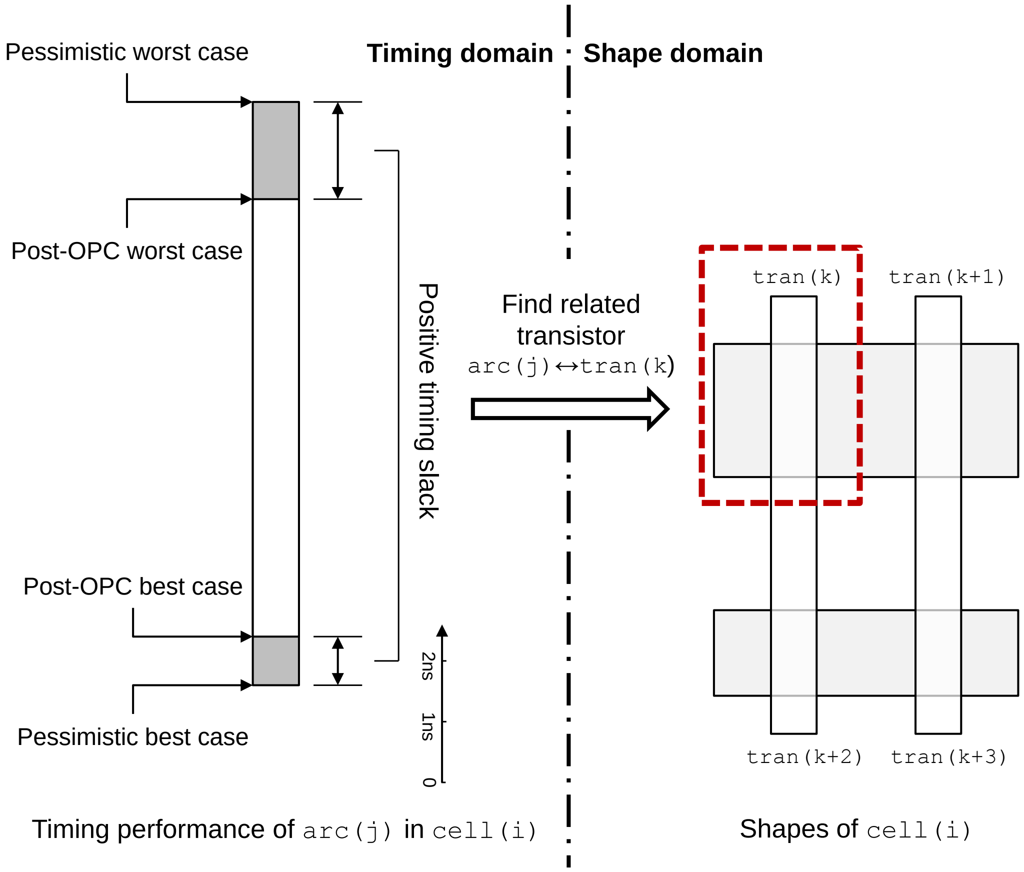


Figure 4.7: Relating timing slack to shape slack

where $D_{0,j}$ is the original delay of the j_{th} timing arc, D_j is the delay when transistor size changes (by varying ΔM and ΔL), $M_k = (W_k)^{-1}$ is the reciprocal of transistor width, W_k, and L_k is the transistor length. SPICE simulations are conducted to characterize the sensitivity coefficients ($\frac{\partial D_j}{\partial M_k}$ and $\frac{\partial D_j}{\partial L_k}$). Results show that the proposed model is accurate over a small variation of M and L with respect to their initially designated values [49]. The relationship between a timing arc `arc(j)` and a transistor `tran(k)` can

```
input  : Original OPCed mask, PVT conditions, SPICE models
output: Simplified mask
foreach cell(i) do
    Initialize "Library Timing" of all timing arcs;
    repeat
        updated ← false;
        Simulate "Post-OPC Timing" of all timing arcs;
        Calculate the slack of each arc (slack vector);
        foreach arc(j) do
            if slack[arc(j)] > $\epsilon$ then
                Find its related transistors;
                foreach tran(k) do
                    Simplify mask layout of tran(k) ;
                    Update mask;
                    updated ← true;
    until updated = false ;
    Output this final mask for cell(i);
```

Figure 4.8: OPC mask simplification algorithm

be expressed as a $m \times n$ matrix $\mathbf{R} = \begin{bmatrix} r_{11} & ... & r_{1n} \\ ... & r_{jk} & ... \\ r_{m1} & ... & r_{mn} \end{bmatrix}$, where

$$r_{jk} = \begin{cases} 1, & \text{if } \left|\frac{\partial D_j}{\partial M_k}\right| > k_M \text{ or } \left|\frac{\partial D_j}{\partial L_k}\right| > k_L, \\ 0, & \text{otherwise.} \end{cases} \tag{4.6}$$

k_M and k_L are threshold values to distinguish related and unrelated pairs, m is the number of arcs, and n is the number of transistors.

The slack vector, $\mathbf{s}$, is a row vector with m columns representing the timing slack (positive or negative) of each arc:

$$\mathbf{s} = [s_1, ..., s_j, ..., s_m], \tag{4.7}$$

where s_j refers to the actual timing slack state of the j_{th} timing arc. It is actually a function of two row vectors $\mathbf{s_{bc}}$ and $\mathbf{s_{wc}}$ representing timing slack at best case and worst case of all timing arcs (best case timing slack can be neglected, considering that only worst case is critical in some designs):

$$\mathbf{s} = H\left(\mathbf{s_{bc}} - \epsilon_{bc}\mathbf{J_{1,m}}\right) \circ H\left(\mathbf{s_{wc}} - \epsilon_{wc}\mathbf{J_{1,m}}\right) \tag{4.8}$$

$$\mathbf{s_{bc}} = \left[s_{bc,1}, s_{bc,2}, ..., s_{bc,m}\right] \tag{4.9}$$

$$\mathbf{s_{wc}} = \left[s_{wc,1}, s_{wc,2}, ..., s_{wc,m}\right], \tag{4.10}$$

where ϵ_{bc} and ϵ_{wc} are the threshold values to leave margins for the timing slacks at best case and worst case. The function H in (4.8) is a Heaviside step function of a matrix which is defined by:

$$H(\mathbf{A})_{ij} \equiv H(\mathbf{A}_{ij}) \tag{4.11}$$

$$H(x) = \begin{cases} 1, & x > 0, \\ 0, & x \leq 0. \end{cases} \tag{4.12}$$

$\mathbf{J_{1,m}}$ is a matrix of ones with a size of $1 \times m$ and is actually an m-column row vector. The operator $\circ$ denotes the Hadamard product for two matrices of the same dimensions. The result of Hadamard product $\mathbf{A} \circ \mathbf{B}$ is a matrix of the same dimensions, which has elements $(\mathbf{A} \circ \mathbf{B})_{ij} \equiv \mathbf{A}_{ij}\mathbf{B}_{ij}$.

The transistor sensitivity vector, $\mathbf{t}$, is defined as:

$$\mathbf{t} = \mathbf{sR}, \tag{4.13}$$

where $\mathbf{t}$ is an n-column row vector, and $t_k = 1$ implies that there is margin for mask simplification for transistor `tran(k)`.

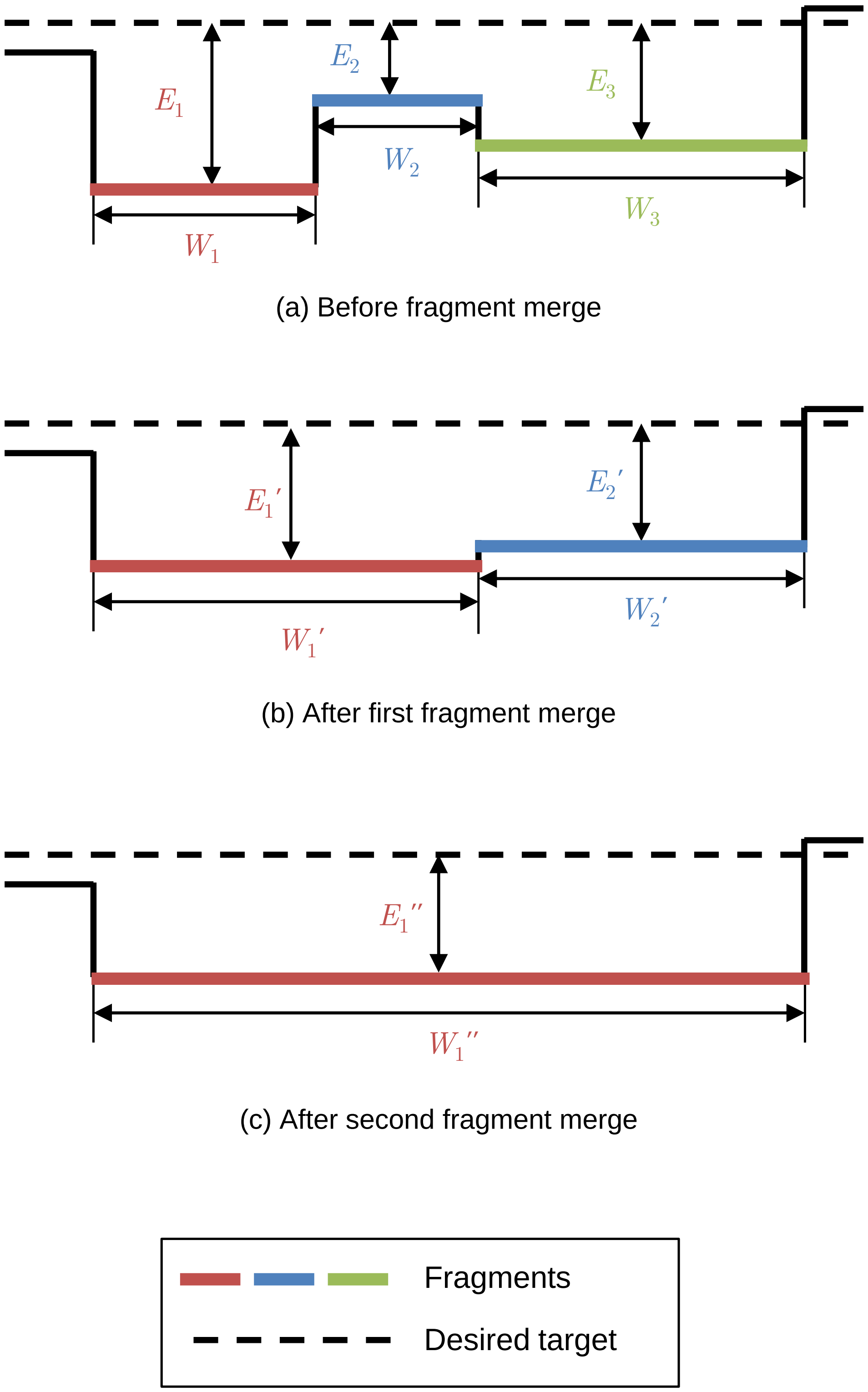
4.4.2 Mask Simplification Algorithm

Figure 4.11(a) shows an example of shifted fragments of a transistor poly mask. Three fragments are highlighted: $F_1\left(W_1, E_1\right)$, $F_2\left(W_2, E_2\right)$ and $F_3\left(W_3, E_3\right)$, where W_i and E_i are the width and edge displacement from the desired target respectively. In the proposed mask simplification algorithm, we seek to align the edges of two consecutive fragments. This will reduce the two vertices of the resulting polygon. We can align the two edges by moving the two edges in opposite direction with each of them covering half the displacement between the edges. This will result in newly formed fragment F_i, with updated W_i and E_i as follows:

$$\begin{cases} W_i{}' & = W_i + W_{i+1} \\ E_i{}' & = \frac{E_i W_i + E_{i+1} W_{i+1}}{W_i + W_{i+1}} \end{cases} . \tag{4.14}$$

The newly aligned edge should fall within the manufacturing grid. This is achieved through a rounding function:

$$\bar{E_i}{}' = Round(E_i{}'). \tag{4.15}$$



(a) Before fragment merge

(b) After first fragment merge

(c) After second fragment merge

Figure 4.9: Definitions of fragment geometry

A cost function is defined to search for a pair of fragments with minimum changes on the layout, *i.e.* two closest fragments with the least truncation error:

$$J_i = \left(\bar{E}_i' - E_i'\right)^2 + \alpha \left(E_i - E_{i+1}\right)^2 \tag{4.16}$$

$$i^* = \arg\min_i J_i \tag{4.17}$$

where α is a non-negative coefficient. The optimal result, i^*, means that two fragments F_{i^*} and F_{i^*+1} are to be merged using (4.14). This simplification is conducted iteratively at both left and right sides of the transistors. The vertex count at each polygon decreases until the transistor sensitivity vector $\mathbf{t} \to 0$ or when the polygon cannot be simplified further, *i.e.* no more fragments can be merged. The achievements in mask cost reduction will be shown in Section 4.5.

Figure 4.11(a-c) shows an example of fragment merge. The fragments F_1 and F_2 in Figure 4.11(a) are merged in the first fragment merge operation. A new fragment, F_1', is formed. F_3 in Figure 4.11(a) is renamed as F_2' in Figure 4.11(b). Another fragment merge in conducted in Figure 4.11(c): F_1' and F_2' are merged to F_1''.

In this work, only fragments in gate region on poly layer are discussed. The remaining region on poly layer is the contact pad and the hammer head. This is maintained from original OPCed mask. Other layers such as diffusion and metal layers are not as critical as poly layer in terms of contribution to timing performance. Therefore, these layers are not discussed in this work and similar works in [14, 24, 49].

4.5 Results and Discussions

To validate this proposed approach, simulations on Nangate 45nm Open Cell Library [41] are conducted. Mentor Graphics Calibre OPCverify and nmOPC [18] are used as lithography simulator and conventional OPC mask generator (for original OPCed mask generation). A 193nm light source with 1.33 hyper-NA immersion lithography is used. The mask simplification algorithm is implemented in Perl scripts and run on a Linux workstation. The PVT corners are all identical to the library (V_{DD}, temperature, mobility and V_{th}), except for the CD variation range. The library assumes a ±10% variation in CD at fast/slow process corners. Multiple lithography simulations are applied and a four-point process window, $[d_{min}, d_{max}] \times [f_{min}, f_{max}]$, is derived, which induces a ±10% variation in CD (same as the library). This process window information will be used in the simulations.

An example on a simple inverter cell (INV_X1) is shown in Section 4.5.1. Circuit level analysis is also discussed in Section 4.5.2.

4.5.1 An Example on Inverter

A standard cell instance INV_X1 (inverter, instance name: U1) in a benchmark circuit c432 from ISCAS'85 [65] is taken for detail explanation. There are two timing arcs and two transistors in U1. The first arc is related with the first transistor, and the second arc is related with the second transistor. This relationship can be express by the **R** matrix in (4.6):

$$\mathbf{R_{U1}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.18}$$

Table 4.4: Mask Simplification Progress (INV_X1)

Iteration No.	$\mathbf{s_{bc}}$[a]	$\mathbf{s_{wc}}$[a]	$\mathbf{s}$	$\mathbf{t}$	# Fragments
0	$[2.14, 8.33]$	$[6.23, 14.69]$	$[1, 1]$	$[1, 1]$	$[11, 11]$
1	$[2.01, 6.78]$	$[5.36, 10.74]$	$[1, 1]$	$[1, 1]$	$[9, 9]$
2	$[1.63, 5.20]$	$[4.76, 7.65]$	$[1, 1]$	$[1, 1]$	$[7, 7]$
3	$[1.39, 4.01]$	$[4.17, \mathbf{3.42}]$	$[1, 0]$	$[1, 0]$	$[5, 5]$
4	$[0.88, 3.99]$	$[3.89, \mathbf{3.36}]$	$[1, 0]$	$[1, 0]$	$[3, 5]$
5	$[\mathbf{0.52}, 4.00]$	$[3.48, \mathbf{3.40}]$	$[0, 0]$	$[0, 0]$	$[2, 5]$

[a] Unit: ps. Bolded numbers are no larger than threshold values.

Table 4.4 shows the information of the progress of the 5-iteration mask simplification loop. Before mask simplification, there is a 2.14ps slack on the best case timing of first arc and 6.23ps slack on the worst case timing of the first arc. This means the timing slack is sufficient for the first transistor to apply a mask simplification. Two fragments are then merged together. Then a timing simulation is conducted again and there is still 2.01ps and 5.36ps left for best and worst cases, which means one more round of simplification can be applied. This is done iteratively until the 5th iteration where only 0.52ps (smaller than the threshold value) is left for the best case timing slack. In this example, the fragment count of the first transistor has already dropped to 2, and this means both the left and right sides of the poly region have been reduced to the simplest version. Therefore, even if the timing slack is sufficient, the iteration will still stop at this situation. Similar operations are applied to the second transistor. Whether a transistor is to be simplified can be known from the transistor sensitivity vector **t** defined in (4.13) and tabulated on the 5th column of Table 4.4. Figure 4.10(a) shows the original OPCed mask layout, and Figure 4.10(b-e) are the intermediate results after each iteration. The output after the 5th iteration shown in Figure 4.10(f) is the final output mask layout.

Fragment count is shown in the last column of Table 4.4. Before the simplification, there are 11 fragments for both transistors' mask layout. After 5 iterations, only 2 fragments are left in the first transistor's mask layout (the lower transistors in Figure 4.10) and 5 fragments are left in the second transistor's mask layout (the upper transistors in Figure 4.10). The overall reduction in terms of vertex count in the gate regions is 68%.

4.5.2 Circuit Level Simulations

The circuit level simulations are based on circuits from ISCAS'85 (c432, c499, c880, c1908, and c2670). The original OPC masks are generated using nmOPC from a pre-established aggressive OPC recipe. The mask simplification procedure is then employed as outlined earlier.

The fragment count and vertex count have been reduced significantly by 46% and 29% respectively using the proposed approach as shown in Figure 4.11 and Table 4.5. In terms of MEBES file size, an average of 37% file size saving has been achieved. Compared to [14], which only take into account circuit performance without exploiting the over-designed time slack, a further reduction of 13-20% have been achieved.

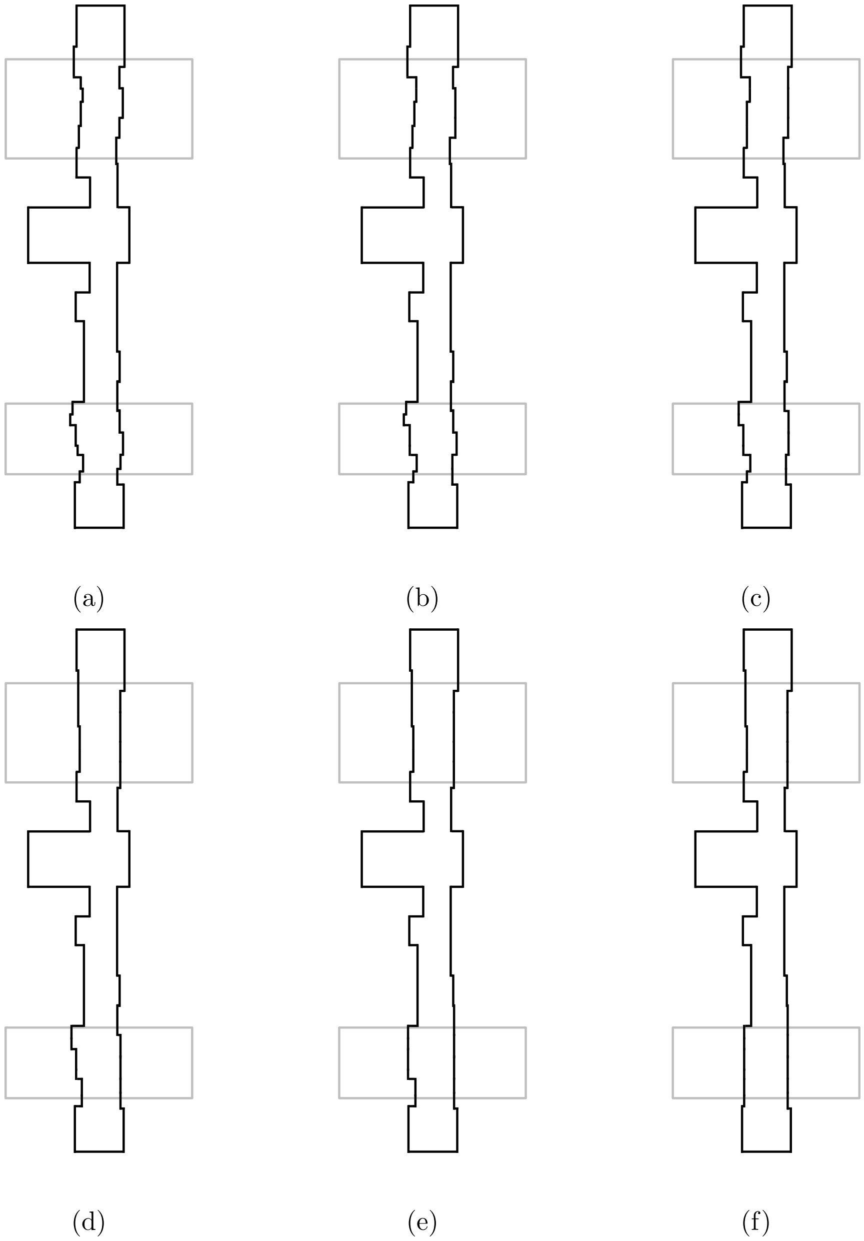
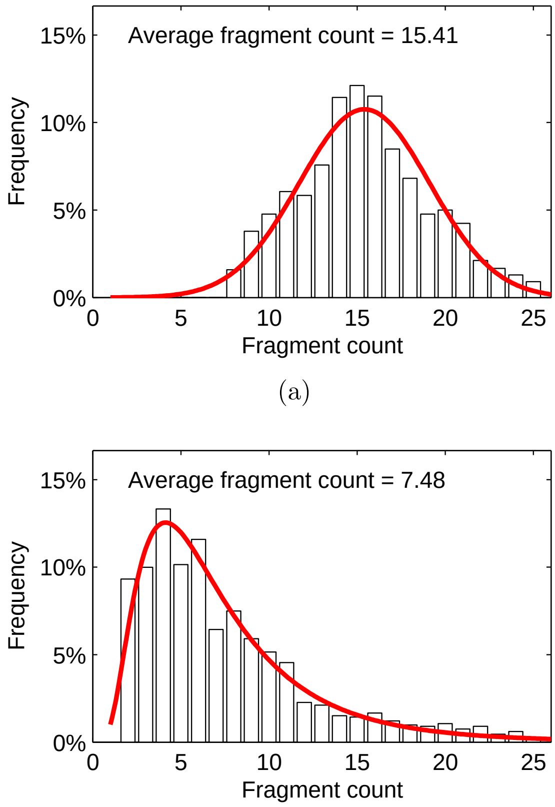


Figure 4.10: Mask simplification progress: (a) Original OPCed mask, (b-f) simplified mask after iteration 1 to 5. (Only OPCed poly layers and non-OPCed diffusion layers are shown.)

To compare the timing slack of the circuits, Monte-Carlo SPICE simulations based on a predefined process, supply voltage range and



(a)

(b)

Figure 4.11: Histogram of fragment count per transistor (c432): (a) original mask, (b) after mask simplification

temperature range (PVT) are conducted to estimate the path delay of the circuit. This is achieved by varying the SPICE model parameters (such as $lint, vth0, k1, u0$, & xj) given in the original Nangate SPICE model. Figure 4.12 and Table 4.6 illustrate the time slack reduction due to mask simplification. Using the original complicated mask, the investigated circuits reported over-design time slack of 10.2% and 24.4% for best and worst case scenarios. With the proposed approach, the masks are simplier with a smaller time slack of 6.7% and 15.4% for best and worst case scenario. This time slack reduction did not result in any timing violation.

Table 4.5: Mean fragment count per transistor and MEBES file size reduction

Circuit	Before	After	Reduction	MEBES reduction
c432	15.4	7.5	51.5%	39.4%
c499	16.1	9.3	42.1%	33.9%
c880	15.5	7.8	49.2%	39.0%
c1908	16.0	8.7	45.7%	35.5%
c2670	15.8	8.9	43.6%	34.8%
Avg.	15.7	8.4	46.3%	36.5%

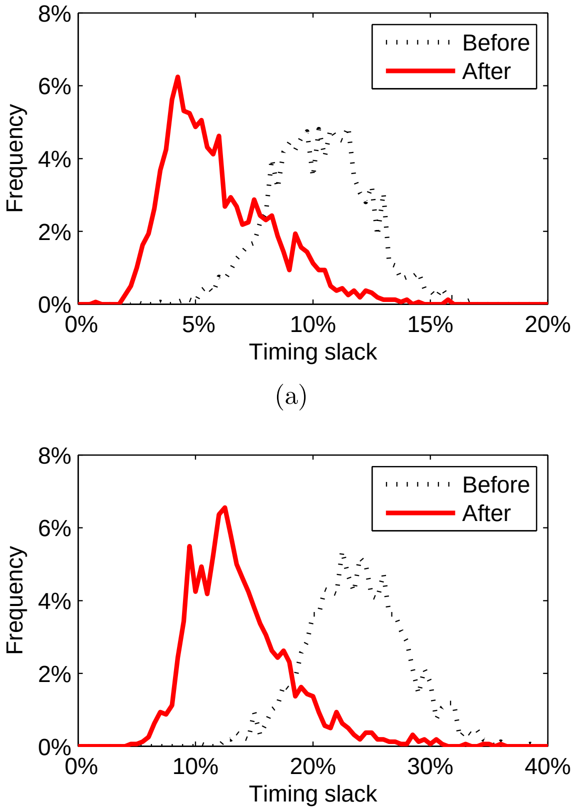
Table 4.6: Timing slack at best cases (BC) and worst cases (WC) before and after applying mask simplification, and its reduction (rd.)

Circuit	BC-Bef.	BC-Aft.	BC-Red.	WC-Bef.	WC-Aft.	WC-rd.
c432	10.4%	6.1%	41.1%	24.0%	14.4%	40.2%
c499	9.2%	6.0%	34.7%	19.9%	13.4%	32.5%
c880	11.8%	7.6%	35.3%	30.2%	16.4%	45.6%
c1908	9.5%	6.3%	33.8%	21.3%	14.6%	31.6%
c2670	10.2%	7.2%	29.6%	26.6%	18.3%	31.2%
Avg.	10.2%	6.7%	34.9%	24.4%	15.4%	36.2%

Run-time is affected by the number of iterations in OPC, n_{OPC}, and number of iterations in mask simplification, n_{Simp}. Run-time (based on the simulation settings) of OPC and mask simplification is tabulated in Table 4.7. On average, the run-time has increased by about 153%. This is due to the extra n_{Simp} loops required to apply mask simplification.

Table 4.7: CPU run-time of OPC and mask simplification, normalized with respect to c432's OPC run-time (172 sec.)

Circuit	# Transistor	OPC	Mask simplification	Increment
c432	626	1.00	1.58	158%
c499	1550	1.86	2.69	145%
c880	1758	1.97	3.01	153%
c1908	2018	2.31	3.64	158%
c2670	4848	3.09	4.71	152%
Avg.				153%



(a)

(b)

Figure 4.12: Histogram of timing slack (c432): (a) best case, (b) worst case

4.6 Conclusion

A mask utility function was proposed to evaluate the OPC masks. Simulations were conducted on a set of benchmark circuits and the result validated this characterization method. With this method, manufacturers would be able to score an OPC mask and choose the mask with optimal utility to save manufacturing cost.

The feasibility of reducing mask cost by exploiting the over-designed timing slacks in digital cells was demonstrated in this chapter. A relationship between time domain and shape domain was found with the help of the timing cost function, which correlates the mask shape with the timing parameter. The reduced timing slack does not incur any timing violation and can reduce the fragment count by up to 46%, which directly translate to mask cost saving. Full circuit analysis has also been conducted to validate the feasibility of this approach on a larger combinational logic.

Chapter 5

Fast Optical Proximity Correction with Timing Optimization Ready Standard Cells

5.1 Introduction

Chapter 2 improves OPC run-time by using a feedback controller. However it is only applicable to stand-alone cells. A application-specific integrated circuit (ASIC) usually consists of a number of standard cells. Run-time issue becomes severe under such circumstances. In this chapter, a fast OPC methodology with timing optimization ready standard cells is designed and implemented. The standard cell layouts are optimized in an off-line process before full chip

OPC is conducted. These timing optimization ready standard cells (TORSC) are stored in a library. In the OPC process, the original cells on the mask are simply substituted with TORSCs. Run-time can be reduced by 5 times with this method since full chip OPC is avoided. Two types of TORSC models are used: one is based on the mask cost saving method proposed in Chapter 2, *i.e.* TPO-TORSC; the other is based on the shape driven method, *i.e.* EPE-TORSC. The proposed fast OPC methodology with two TORSC models achieves two different design objectives. TPO-TORSC reduced mask cost with good timing performance matching, and EPE-TORSC attains best timing performance matching with penalty in mask cost. Since mask cost is a desirable metric in OPC design [9], a hybrid approach is proposed to combine the advantage of TPO-TORSC and EPE-TORSC. Simulation result show a good timing accuracy with little penalty on mask cost.

The rest of this chapter is organized as follows. In Section 5.2, an overview on existing electrically driven OPC methods is presented. Section 5.3 introduces the fast OPC method with TORSC. Section 5.4 shows the preliminary simulation results. Section 5.5 demonstrates the feasibility and effectiveness of a hybrid approach. Finally, conclusions are drawn in Section 5.6

5.2 Existing Electrically Driven OPC Methodologies

Electrically driven OPC methods place emphasis on the electrical performances rather than fidelity of printed shapes. These methods aim at reducing the deviation of electrical performance: drive and leakage current, timing, power, etc. [14, 21, 22] However, existing electrically driven OPCs are mostly full chip based. In order to achieve the electrical performance for each transistor on the entire layout, computational time will be extremely expensive. As illustrated in Section 1.3, run-time rises exponentially when the transistor count increases due to the time complexity of lithography simulation. To address the issue of rising run-time, a cell-wise OPC scheme is introduced in this chapter to save computational effort. The idea of cell-wise OPC is to save run-time by identifying standard cells on the mask and substituting them from the OPC-ready cell library generated off-line [42, 71–74]. This idea can be employed to reduce the computational effort of those full chip based electrically driven OPC. Moreover, existing electrically driven OPC schemes only focus on transistor level electrical behaviors, especially drive and leakage current. Matching in these parameters does not guarantee timing performance. For digital circuits, gate level path delay is a more desirable performance parameter and is adopted in the proposed fast OPC methodology.

5.3 Fast OPC Methodology

5.3.1 OPC Flow

Existing electrically driven schemes usually have to be subjected to full chip OPC process. A typical flow is to iterate until all transistors and/or cells have met the local electrical goal. And during each iteration, the mask is corrected based on a certain mask generation algorithm. Figure 5.1(a) illustrates this typical flow.

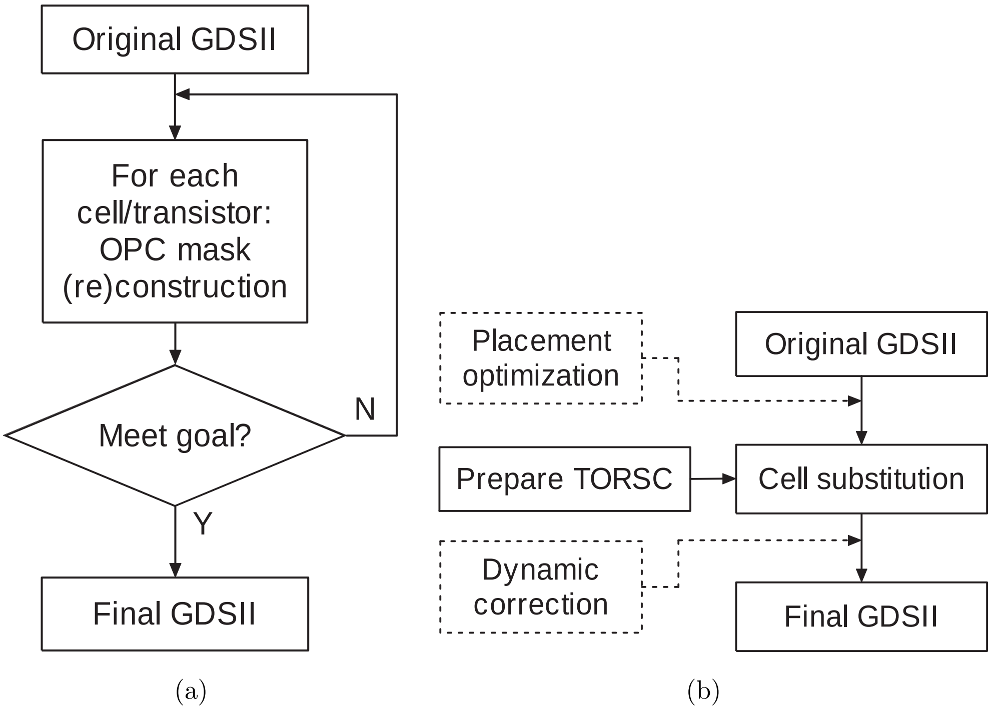


Figure 5.1: Flowcharts of OPC schemes: (a) existing electrically driven OPC; (b) proposed fast OPC methodology

In the proposed fast OPC methodology, the iteration is done off-line in the production of TORSC. Once the TORSC library has been built, simple cell substitution can be employed for full chip OPC. Optional placement optimization and dynamic corrections can be taken to further improve the overall electrical performance. The flowchart of this methodology is shown in Figure 5.1(b). By using such flow, full chip OPC with many iterations can be avoided. Therefore, the computational effort is reduced significantly compared to full chip OPC.

5.3.2 Timing Optimization Ready Standard Cells

The off-line TORSC generation flowchart is shown in Figure 5.2. For each standard cell, the original mask is subjected to iterations. It loops until the cell level electrical goal is achieved.

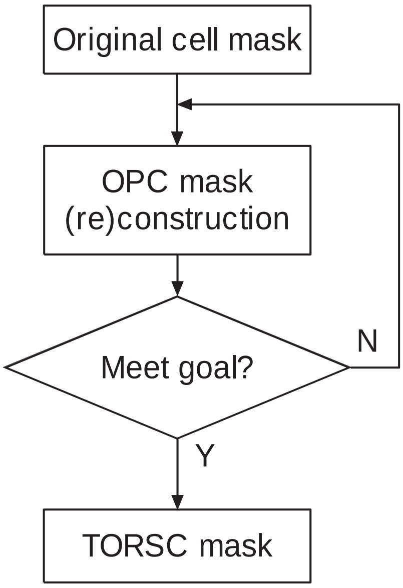


Figure 5.2: TORSC generation flow

A model which aims at minimizing absolute deviations of timing performances can be employed to generate a TORSC library. In this work, TPO-OPC based on method in Chapter 2 with emphasis on gate delay rather than drive and leakage current is employed. For comparison, TORSC based on EPE-OPC similar to the method in Ref. [52] has also been included. They are referred as TPO-TORSC and EPE-TORSC in this chapter respectively.

5.3.2.1 TPO-TORSC

The target of TPO-TORSC is to minimize the absolute timing performance error, $|TPE|$, as described in (2.1). The procedure to generate a TPO-TORSC mask of a standard cell is described below. At first, T_{design} is extracted from the given standard cell, before the TPO-TORSC flow starts. In each iteration, the OPC engine first reconstructs the mask based on $|TPE|$ of the previous iteration. After that, lithography simulation is applied and the printed shapes are extracted. Then a SPICE simulation is performed and $|TPE|$ of the current iteration is calculated. If $|TPE|$ of the current iteration meets the timing requirement (for example, $< 2\%$), the OPC engine exits the loop and output the mask in this final iteration.

In this chapter, worst case propagation delay is selected as the timing performance. Therefore, for each cell, the worst case propagation delay (both t_{pHL} and t_{pLH}) is optimized. However, the propagation delays for non-worst cases are not optimized.

5.3.2.2 EPE-TORSC

The EPE-TORSC masks are generated using model based OPC method. This method targets at minimizing edge placement error (EPE). Idea of this model is from Ref. [52]. It is employed and modified to optimize for each standard cell separately. It should be mentioned that EPE-TORSC is not only shape driven but also electrically driven. Hence, all cases (worst or non-worst) of delays of an EPE-TORSC cell are optimized. Therefore, the individual cell of EPE-TORSC outperforms that of TPO-TORSC in terms of timing accuracy.

5.4 Preliminary Results

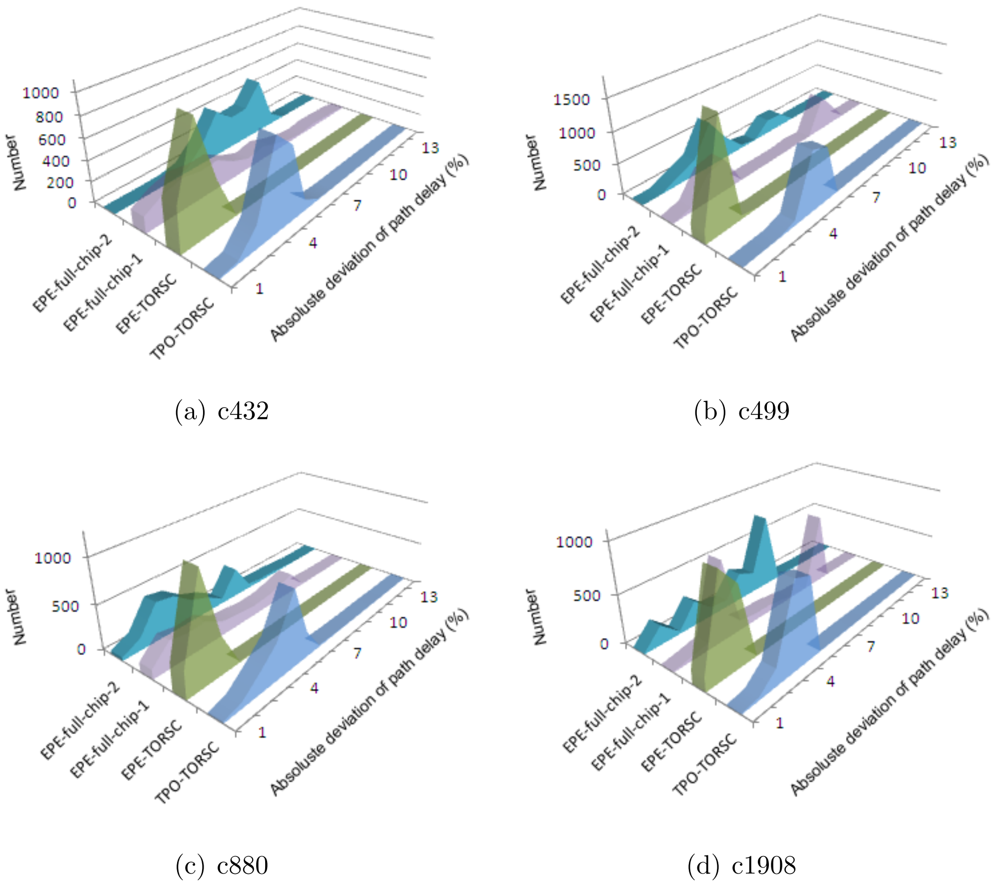
The simulations are based on Nangate 45nm Open Cell Library [41]. Similar simulation settings as Chapter 2 are adopted. TPO-TORSC and EPE-TORSC are generated beforehand (both with a same group of 28 standard cells). Four test circuits (c432, c499, c880, & c1908) from ISCAS'85 benchmark [65] are synthesized with the 28 standard cells. The four test circuits are then placed and routed with two sets of P&R specifications. This results in two significantly different post layout GDS: GDS-1 and GDS-2.

Four OPC schemes are applied to the GDS-1 and GDS-2 respectively. OPC scheme (1) is the proposed method with TPO-TORSC. OPC scheme (2) is with EPE-TORSC. OPC scheme (3) and (4) are full chip shape driven OPC methods with conventional EPE algorithm. [1]

Path delay and mask size are compared. Path delay refers to the propagation time from input port to output port of a full chip. For each test circuit, a number of 2,000 random input patterns are excited into the input ports, and the rising or falling edge is captured on the specific output port to measure path delay. The measured path delays are compared to nominal path delays [2] of the circuits. 3D area plots of the distribution of these resulting full chip path delay $|TPE|$ (absolute deviation from design value)

[1] OPC scheme (3) and (4) have different EPE-OPC settings.

[2] Nominal delay refers to the delay with originally designed transistor sizes.



(a) c432

(b) c499

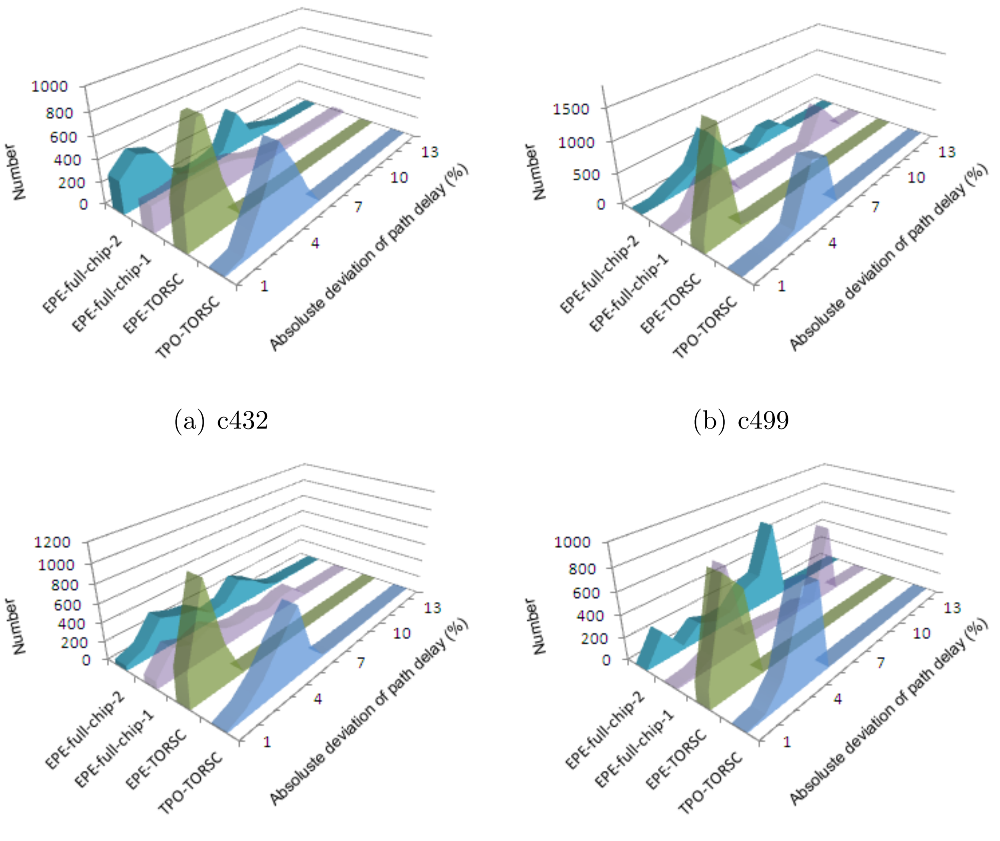
(c) c880

(d) c1908

Figure 5.3: Full chip path delay $|TPE|$ distribution, GDS-1 (P&R method 1)

are then plotted in Figure 5.3 and 5.4, for GDS-1 and GDS-2 respectively. The X-axis is the full chip path delay $|TPE|$ in percentage point (from 1% to 15%), the Y-axis is different type of OPC schemes, and the Z-axis is the number that corresponding to that particular absolute deviation.

To evaluate the mask size (or mask complexity), a method to count the number of vertices is employed [23]. Vertex count implies mask complexity and is proportional to mask write time. The vertex count of each mask is normalized with respect to the vertex count of the diffusion layer mask of circuit c432, and they are listed in Table 5.1. Only masks of GDS-1 are



(a) c432

(b) c499

(c) c880

(d) c1908

Figure 5.4: Full chip path delay $|TPE|$ distribution, GDS-2 (P&R method 2)

tested, since masks of GDS-2 are having almost the same mask complexity.

Since this method does not apply placement optimization and/or dynamic correction like other cell-wise OPC schemes, the overall path delay may deviate significantly from designed value even though local timing performance matching is guaranteed. From Figure 5.3 and 5.4, it seems that the deviation due to different placements of TORSC is not significant. To further validate this observation, investigation is conducted into the timing performance matching for the TPO-TORSC on 8 different GDS layouts with different P & R settings. Both average and standard deviations on the timing performance matching are shown in Figure 5.5. As illustrated, there is very little difference between the different GDS layouts.

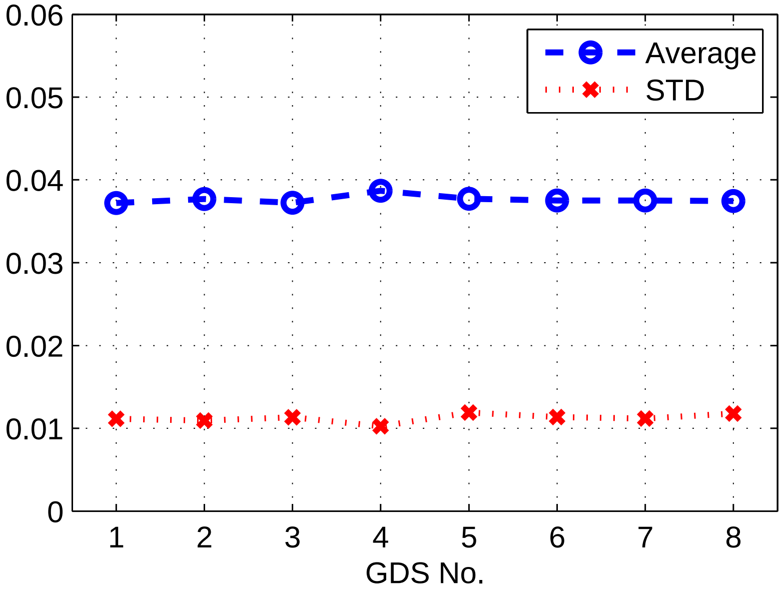


Figure 5.5: Simulation of 8 different GDS's

Results in Figure 5.3 and 5.4 show that the proposed OPC methodology with TORSC models (TPO-TORSC & EPE-TORSC) outperforms full chip methods (EPE-full-chip-1 & -2) in terms of timing accuracy in two aspects: the mean absolute deviation is lower (0.037-0.051 for TPO-TORSC and 0.015-0.019 for EPE-TORSC), and the spread of deviation is also much smaller (0.013 and 0.010). In addition, EPE-TORSC is about 2.5 times more accurate than TPO-TORSC. The reason is explained earlier in Section 5.3.2. EPE-TORSC ensures timing accuracy in all cases of delays while TPO-TORSC only optimizes worst case propagation delay. Path delay does not necessarily excites worst case transitions on all cells, since it consists of not only worst case but also non-worst cases of individual cells. Therefore the advantage of EPE-TORSC in terms of individual cell timing accuracy results in the better full path timing accuracy.

Table 5.1: Normalized mask vertex count w.r.t c432 diffusion design mask vertex count, Scheme (1): TPO-TORSC; (2): EPE-TORSC; (3): EPE-full-chip-1; (4): EPE-full-chip-2

Layer	Scheme	c432	c499	c880	c1908	Multiple
	Design	1.00	1.95	3.12	2.79	1×
	(1)	1.00	1.95	3.12	2.79	1.00×
Diff	(2)	2.50	5.07	8.72	6.66	2.59×
	(3)	4.31	7.80	12.54	11.23	4.05×
	(4)	5.54	16.28	25.25	22.88	7.90×
	Design	2.24	4.67	6.30	6.53	2.23×
	(1)	2.24	5.57	6.30	7.30	2.42×
Poly	(2)	5.10	12.02	14.45	14.84	5.24×
	(3)	4.40	20.04	17.48	23.73	7.42×
	(4)	8.44	27.97	27.10	33.83	11.00×

Improvements in OPC run-time are shown in Table 5.2. On average, run-time speed up of 3-8 times is achieved. Run-time of proposed method with EPE-TORSC is identical to that of TPO-TORSC. As compared to the best previous fast OPC method in [71], which took 0.17 seconds per cell, the proposed method in this work (0.03 seconds per cell) still achieves obvious improvements. The run-time cost to generate the 28-cell TORSC library is 43 minutes for TPO-TORSC, and 82 minutes for EPE-TORSC. However, this off-line run-time is negligible if the chip is large (the test circuits in this work are tiny compared to industry circuits). The entire above mentioned run-times are measured on a same Linux machine.

It can be further concluded from Table 5.1 that, purely electrically driven method (TPO-TORSC) usually results in smaller mask sizes than shape driven

Table 5.2: OPC run-time and speedup, (1): TPO-TORSC; (3): EPE-full-chip-1; (4): EPE-full-chip-2

	(1)	(3)	(4)	(3)/(1)	(4)/(1)
c432	5s	15s	37s	3.00×	7.40×
c499	7s	58s	86s	8.29×	12.29×
c880	12s	56s	110s	4.67×	9.17×
c1908	10s	82s	132s	8.20×	13.20×
Speedup				6.04X	10.51X

methods (EPE-TORSC, EPE-full-chip-1 & -2). TPO-TORSC model is based on simple rectangular tuning and thus results in significant mask cost saving. However, its poorer timing performance matching than EPE-TORSC model might results in sub-optimal digital circuit performance, especially along the critical timing path. On the other hand, although EPE-TORSC gives the best timing performance matching, the mask cost might becomes prohibitively large especially for very large digital design. Closer examination by Ref. [14] reveals that accurate timing performance matching is only required for critical path delay which determines the worst case setup and hold time. Therefore, a hybrid approach is proposed which applies EPE-TORSC for digital cells along the critical path and TPO-TORSC on the rest of the circuit. This approach allows best timing performance matching without incurring huge mask cost.

5.5 Hybrid Approach

As mentioned earlier, a hybrid approach might allow better trade-off between mask sizing and timing performance matching. A previous work [75] on critical path identification enables this work to implement this proposed approach.

Figure 5.6 shows an example of identifying critical path and placing cells. In the digital circuit block, the red-colored cells are marked as critical cells, as they appear on the critical path of the block.

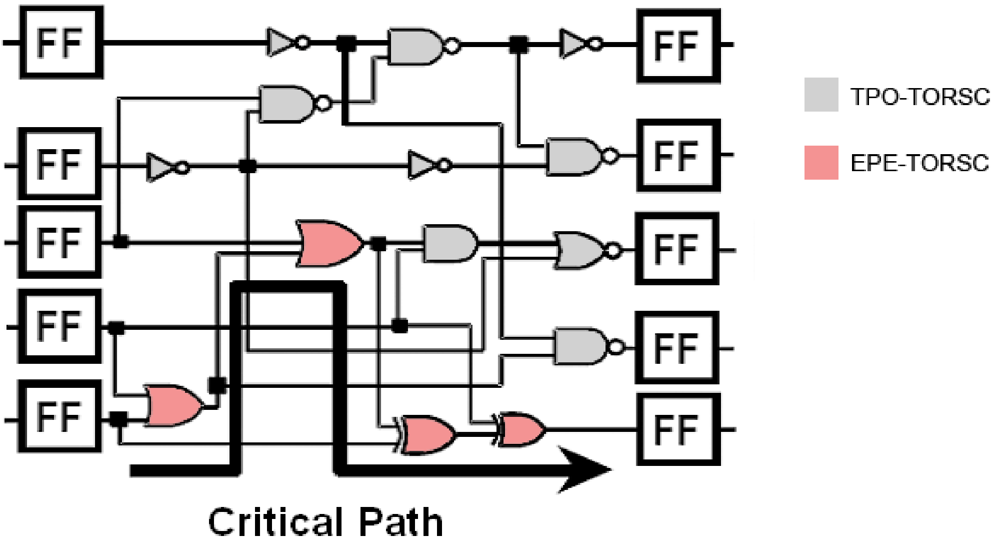


Figure 5.6: Example in a digital circuit block

5.5.1 Flow of Proposed Hybrid Approach

The proposed OPC flow with TORSC models is modified. Based on the flowchart in Figure 5.1(b), two libraries (TPO-TORSC and EPE-TORSC), instead of one library, should be prepared before applying OPC. In addition, information of critical paths (cells on critical path, input and output ports, and input vectors to excite the critical paths' delays) should be gathered and passed to the OPC engine. Industrial tools such as Synopsys PrimeTime [76] can be employed to conduct accurate static timing analysis (STA) and estimate critical paths. Figure 5.7 illustrates the flowchart of this critical path based hybrid approach. When applying OPC, the engine need to know if the cell is on a critical path, and select the appropriate optimized cell layout from the corresponding libraries.

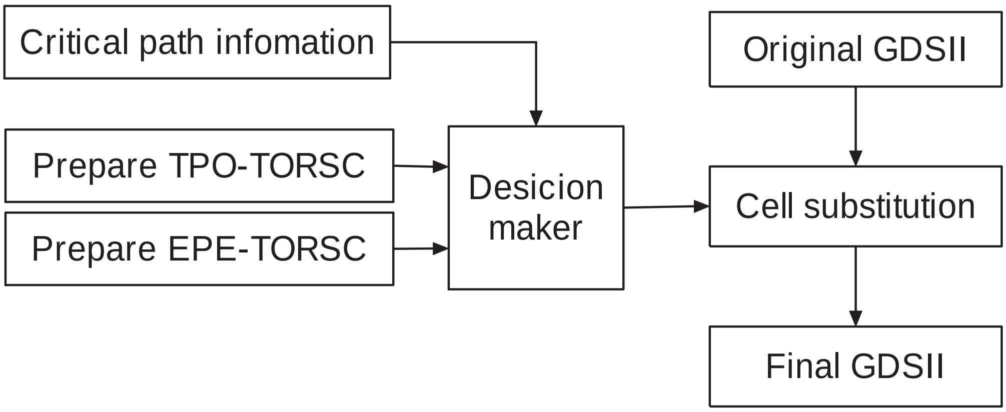


Figure 5.7: Flowchart of proposed hybrid approach

5.5.2 Simulation Results

A simulation set is conducted to evaluate the performance of the hybrid approach. Results of the hybrid approach are compared to pure TPO-TORSC and pure EPE-TORSC methods. Same simulation settings as in Section 5.4 are applied. Considering the scale of these four benchmark circuits, 100 most critical paths are picked using Synopsys PrimeTime, and the occurrence of each cell is counted. The concept of "critical cells" are defined as the cells that amounts up to top $P\%$ of the overall occurrence, where P is usually from 90 to 100. In this work, $P = 90$ (Hybrid 1) and $P = 100$ (Hybrid 2) are used and compared. In Table 5.3, the numbers of critical cells are counted and their corresponding proportions are calculated. The smaller the proportion is, the more possibility to have mask cost reduction, since more TPO-TORSC rather than EPE-TORSC will be used.

For smaller circuits, the proportions of critical cells are large since critical paths are likely to pass through more cells. Therefore, larger circuits are expected to have greater reduction in mask cost.

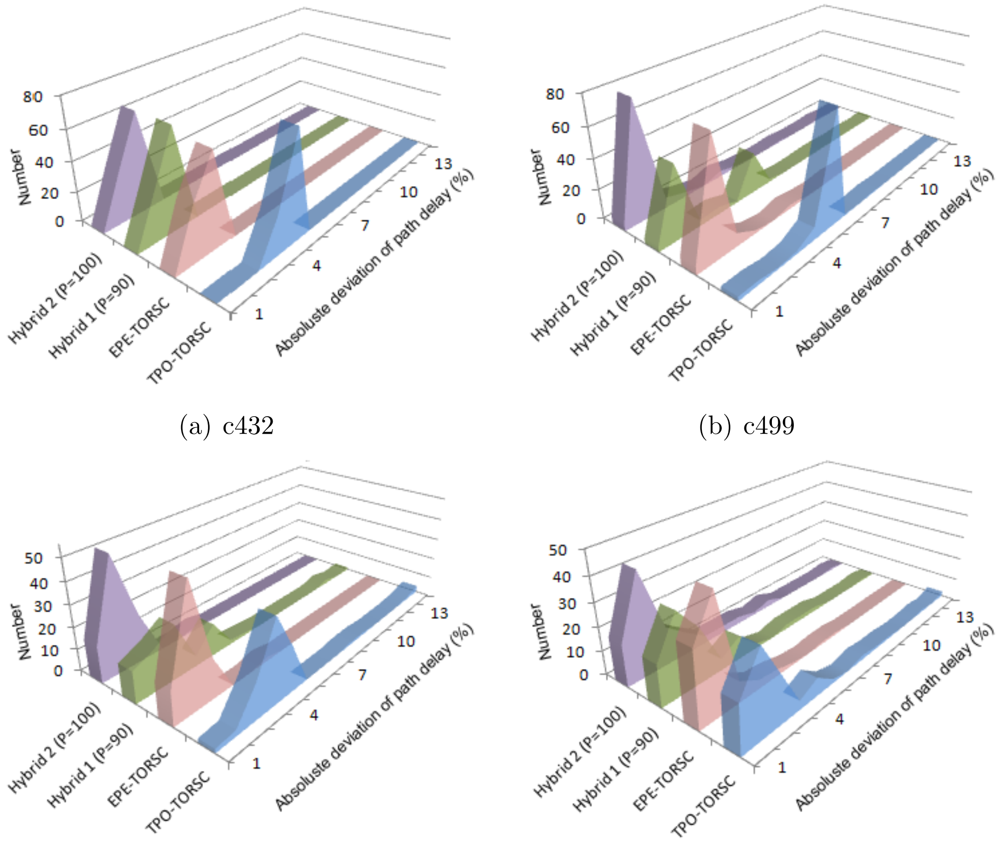
Table 5.3: Number of critical cells

Circuit	c432	c499	c880	c1908
Total # of cells	122	224	361	314
Hybrid 1 critical cell	63 (52%)	158 (71%)	70 (19%)	73 (23%)
Hybrid 2 critical cell	103 (84%)	206 (92%)	146 (40%)	116 (37%)

In order to make a fair comparison, two sets of simulations are conducted. The first set is to test on the 100 critical paths. The second set is to test on 500 random collected input vectors (similar to Section 5.4). For both sets, full chip path delays are measured, and they are compared to the nominal path delays. Figure 5.8 shows the absolute deviation of path delays of the first set, and Figure 5.9 shows those of the second set. Mask sizes in terms of vertex count are listed in Table 5.4.

The following can be derived from these results:

- For simulation set 1 in Figure 5.8, Hybrid 2 almost perform as well as EPE-TORSC. Hybrid 1 is between EPE-TORSC and TPO-TORSC.
- For simulation set 2 in Figure 5.9, Hybrid 2 also achieves good accuracy as EPE-TORSC. On the other hand, Hybrid 1 follows more closely to TPO-TORSC, especially on larger circuits. However, deviations on random paths are not significant since the overall path



(a) c432

(b) c499

(c) c880

(d) c1908

Figure 5.8: Simulation set 1: critical paths

delay on random paths is shorter than that on critical paths and there is more timing slack.

- There is a trend that the similarity between hybrid approach and EPE-TORSC depends on the proportion of critical cells. For smaller circuits, the total number of cells is small. For a cell on a small circuit, its probability to be a critical cell is greater than a cell on a larger circuit. Therefore, the proportion of critical cells for smaller circuits is greater than larger circuits, and ybrid approaches follow EPE-TORSC rather than TPO-TORSC. For designers, if they wish a circuit to have

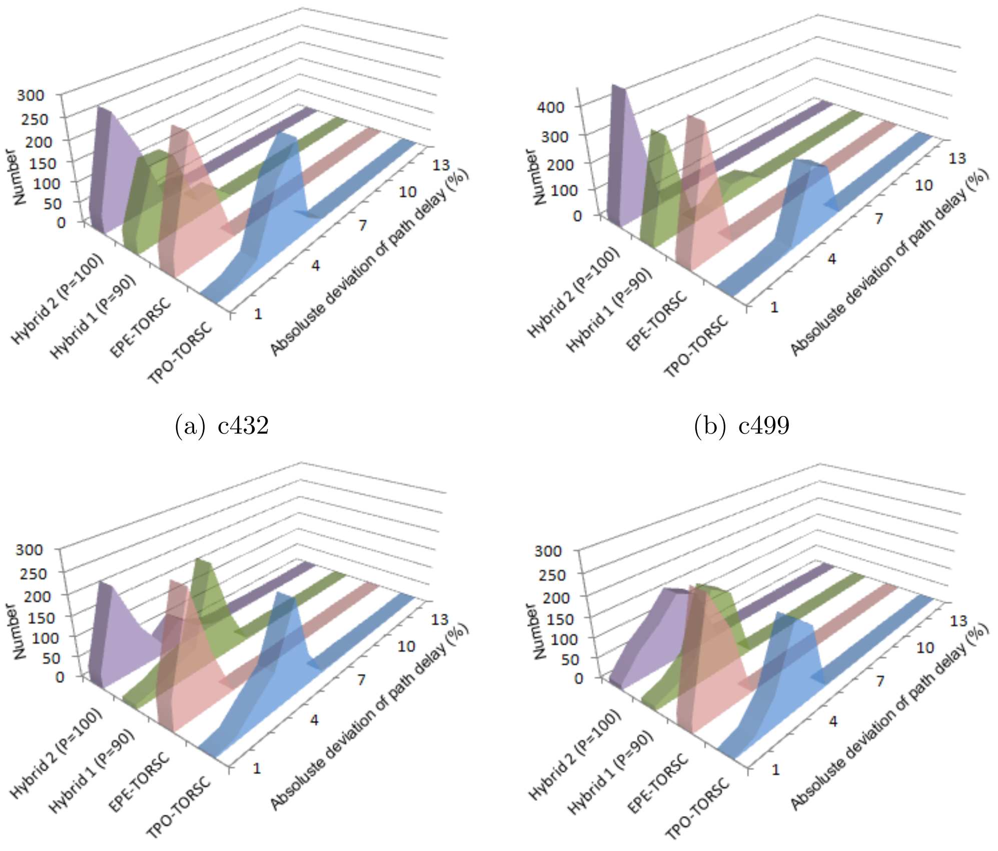
Table 5.4: Normalized mask vertex count w.r.t c432 diffusion design mask vertex count, Scheme (1): TPO-TORSC; (2): EPE-TORSC; (5): Hybrid 1 (P=90); (6): Hybrid 2 (P=100)

Layer	Scheme	c432	c499	c880	c1908	Multiple
Diff	Design	1.00	1.95	3.12	2.79	1×
	(1)	1.00	1.95	3.12	2.79	1.00×
	(2)	2.50	5.07	8.72	6.66	2.59×
	(5)	1.92	4.20	4.08	3.78	1.58×
	(6)	2.22	4.88	5.19	4.54	1.90×
Poly	Design	2.24	4.67	6.30	6.53	2.23×
	(1)	2.24	5.57	6.30	7.30	2.42×
	(2)	5.10	12.02	14.45	14.84	5.24×
	(5)	3.74	10.14	7.84	9.52	3.53×
	(6)	4.44	11.56	9.51	10.96	4.12×

better timing performance, they might need to manually increase the probability of a cell to be critical cell. An easy way to implement this is to report more critical paths from STA tools.

- Run-time of the hybrid approach is identical to TPO-TORSC in Table 5.2. However, the run-time to collect critical path information should be considered.
- Mask cost reduction from EPE-TORSC in terms of mask vertex count is, for Hybrid 1: 39% on diffusion layer and 33% on poly layer, for Hybrid 2: 27% on diffusion layer and 21% on poly layer. More reduction from full chip OPC recipes is expected, since mask size of EPE-TORSC method is already smaller than full chip OPC recipes (in Table 5.1).

To sum up, the proposed hybrid approaches achieve satisfactory timing accuracy. Hybrid 2 with $P = 100$ performs almost as well as EPE-TORSC on



(a) c432

(b) c499

(c) c880

(d) c1908

Figure 5.9: Simulation set 2: random paths

critical paths. Both hybrid approaches reduce mask vertex count reasonably, and this directly reduces mask write time and mask cost.

5.6 Conclusion

A fast OPC methodology with timing optimization ready standard cells (TORSC) is proposed and implemented. This methodology bridges from electrically driven OPC to cell-wise OPC. The major advantages over existing electrically driven OPC and conventional EPE-OPC are saving in computational effort, better timing accuracy and mask cost reduction. This chapter employs TPO-TORSC and EPE-TORSC models, and implements them under the proposed flow. Other models using electrically driven OPC engines can also be integrated with this proposed methodology [14, 21, 22]. In order to further reduce mask cost without too much sacrifice on timing accuracy, a critical path based hybrid approach is proposed and implemented. Satisfactory timing accuracy and mask cost reduction is achieved.

Chapter 6

Conclusion

6.1 Summary

This thesis explored the topic of semiconductor design for manufacturing (DFM) and focused on the issues of optical proximity correction (OPC). New techniques are proposed to resolve problems in mask cost, circuit performance, convergence and run-time.

In Chapter 2, a timing performance oriented OPC approach is proposed. The problem is formulated into a feedback control framework with the circuit's timing performance such as propagation delay feedbacked to the OPC engine. The result outperforms conventional OPC schemes, specifically, 30% reduction in mask size and 5% improvement in timing accuracy is achieved. In addition, the use of feedback control theory is adopted in this work. A proportional-integral (PI) controller for generating OPC masks is designed and implemented. An iterative feedback tuning (IFT) method is also employed to re-tune controller parameters when environments change. The advantage of this approach is that it does not require prior modeling of the plant. The tuning could be applied online while the system is running in a closed loop. Simulation results show that improvement in convergence time is achieved: the number of iterations is reduced by 80%.

In Chapter 3, a process window aware OPC technique is proposed and implemented. The process window is considered in the algorithm. The retargeting process before applying OPC is also optimized. Timing characteristics are employed as a direct metric for the retargeting process. Simulation results demonstrate the feasibility of the proposed approach in terms of timing accuracy, process window, and mask cost. The implementation of timing characteristics as direct metric enables the algorithm to achieve better timing accuracy (improve by 2-5%) compared to other electrically driven OPC techniques. Due to this accurate timing performance, the observed process window for the benchmark circuits is enlarged by 88% compared to previous methods. This directly translates to greater robustness against process variations. The aggressiveness of OPC is also reduced: a 73% mask reduction in terms of mask fragment count is achieved.

In Chapter 4, a methodology is designed and implemented to simplify OPC mask using over-designed timing slack. First, an approach to characterize the relationship between timing yield and OPC mask cost is proposed. Through a mask utility function, optimal OPC schemes can be chosen to obtain low mask cost and good timing yield. This motivates this work to conduct mask simplification which incurs little penalty on timing yield. The proposed OPC mask simplification method is compatible with any existing OPC schemes. The over-designed timing slack can be extracted from the difference between post-OPC simulations and library data. A fragment merge algorithm is proposed in this work to reduce the number of fragments in the OPC masks. Simulation results on standard cells show a 51% reduction in terms of polygon vertex count, which directly relates with a significant reduction in mask cost. Timing closure is guaranteed and no changes will be made on the designed logic.

In Chapter 5, a fast OPC methodology based on cell-wise optimization is proposed and implemented. The full layout is split into multiple single cells and OPC is conducted in parallel, for each type of standard cell. The standard cells after a timing performance oriented OPC, *i.e.* timing optimization ready standard cells (TORSCs), are stored in a lookup table. In the last step of OPC process, original layouts are substituted with the TORSCs. Since full chip OPC is avoided, simulation results when compared to conventional OPC approaches in the literature demonstrate the reduction in run-time. Depending on the circuit test-set, an average run-time improvement between 3 to 8 times is achieved for circuit size with 100 - 400 cells. Further improvements in terms of balance between timing accuracy and mask cost can be obtained by adopting a hybrid approach by only optimizing the timing performance of critical paths.

6.2 Future Work

The theme of this thesis thus far is the issues of mask cost, circuit performance and convergence in OPC. Much remains to be done in the realm of semiconductor design for manufacturing. Possible future work includes OPC methods for double patterning, extreme ultraviolet lithography, and integration of resist processing into the OPC framework.

6.2.1 OPC for Double Patterning Techniques

Double patterning (DP) process is referred as a patterning process where an etch step (or some other image preserving technique) occurs between two consecutive exposure steps to form the intended design pattern on silicon wafer. Such process enables pitch relaxation and effectively allows processing with k_1 factors smaller than the theoretical Rayleigh limit of 0.25. This requires decomposition of the intended design pattern in two parts by splitting it relative to the densest pattern pitches [77].

Although masks are simply decomposed into two separate masks as noted in Figure 6.1, OPC for these two masks is not a straightforward task. Critical line-end control is required to ensure overlap or avoid bridging risk [78]. Therefore, improvements in single exposure OPC line-end modeling and line-end OPC correction are likely required with double patterning. Also required is overlap aware OPC to ensure accurate creation of the combined final etched patterns through process and overlay errors.

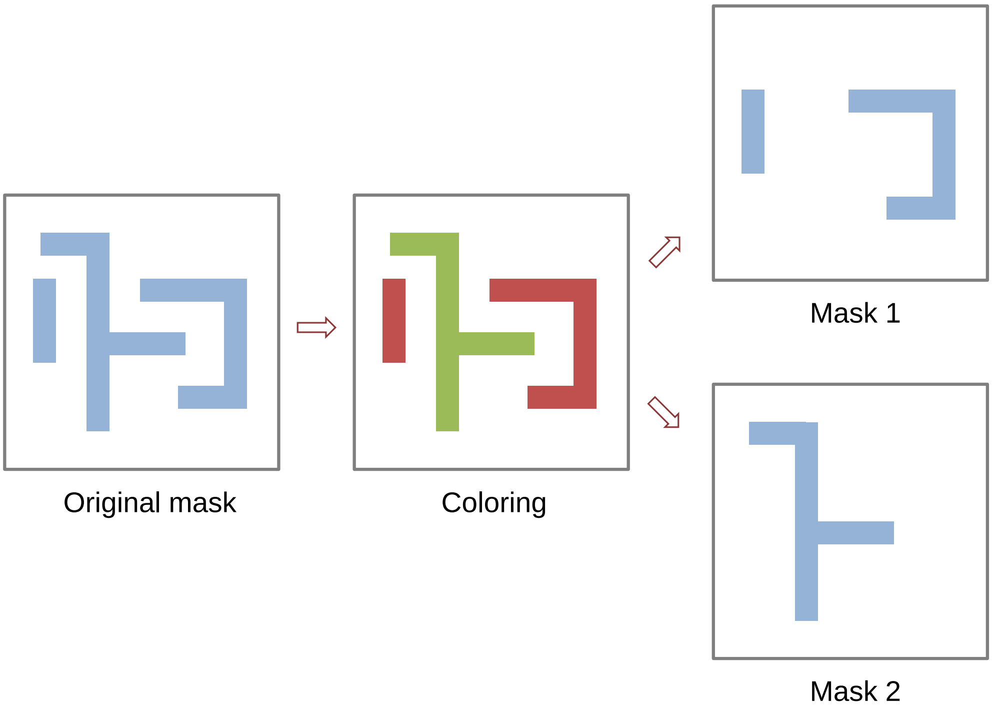


Figure 6.1: Mask decomposition of double patterning [5]

6.2.2 OPC for Extreme Ultraviolet Lithography

Extreme ultraviolet (EUV) lithography utilizes a light source with an extreme ultraviolet wavelength at 13.5 nm [79]. Although the k_1 factor is large for EUV lithography compared to deep ultraviolet (DUV) lithography, OPC is still required to print the intended patterns on the wafer. This is primarily because of new non-idealities, related to the inability of materials to absorb, reflect, or refract light well at 13.5nm, which must be corrected by OPC [80]. For EUV, OPC is much more than conventional optical proximity correction. Issues such as circuit performance and convergence still exist in EUV lithography. Research remains to be conducted to fill these gaps in OPC for EUV lithography. Possible difficulties are the flare effects (the flare value assumed in OPC image calculation is incorrect) and horizontal-vertical biasing (the bias applied during OPC to account for the print difference between horizontal and vertical lines is incorrect). The modeling of EUV lithography is still a complicated work. The iterative feedback tuning method used in Chapter 2 can be incorporated into OPC schemes for EUV.

6.2.3 Integration of Resist Processing in OPC

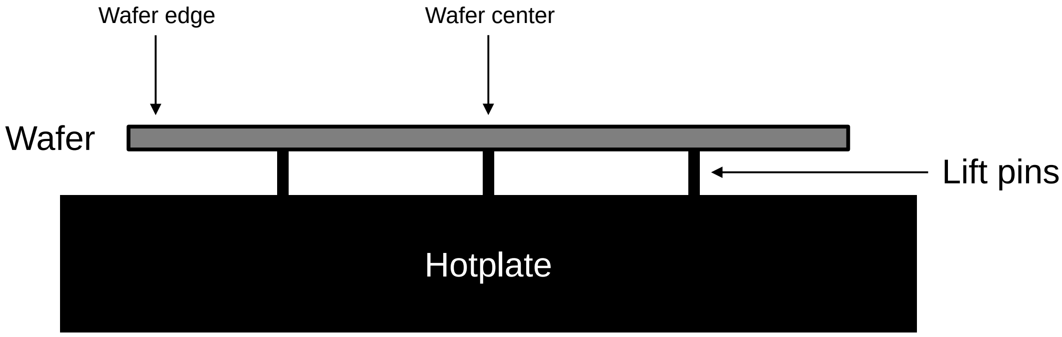


Figure 6.2: PEB: The temperature at the center of the hotplate increases more slowly than the temperature at the edge [79].

The simulations of this thesis only focused on exposure process in lithography. However, apart from the exposure step, the resist processing step (post exposure bake, or PEB) in lithography is also important. [1] It is desirable to bring the resist processing part into the OPC framework, in order to achieve more accurate simulation results. Modern chemically amplified resist (CAR) is sensitive to PEB time, temperature, and delay, as most of the "exposure" reaction of such resist actually occurs in the PEB [79].

[1] The overall lithography flow is shown in Figure 1.1.

Figure 6.2 shows a typical PEB hotplate configuration [81]. During PEB, temperature variation across the hotplate is often observed during thermal ramps. Experimental results demonstrated that this thermal non-uniformity causes CD variation across the wafer. Our research group has been looking into both OPC and the baking process in lithography in recent years [22, 40, 82–84]. This would be a good time to merge the research outcomes of the two parts. Circuit performance varies at different locations on the wafer. Therefore, it is of great interest to integrate the resist processing in OPC with considerations of thermal non-uniformity, and to reduce the intra- and inter-wafer deviations in circuit performance.

Author’s Publications

The author has contributed to the following publications:

Peer-reviewed Journal Papers

[1] **Y. Qu**, C. H. Heng, A. Tay and T. H. Lee, ‘Characterizing of timing yield and manufacturing cost for optical proximity correction masks,” manuscript submitted to *Journal Of Vacuum Science And Technology B*, 2013.

[2] **Y. Qu**, C. H. Heng, A. Tay and T. H. Lee, “OPC mask simplification using over-designed timing slack from post simulation,” manuscript submitted to *Electronic Letters*, 2013.

[3] **Y. Qu**, C. H. Heng, A. Tay and T. H. Lee, “Optical proximity correction for accurate timing performance, process window enlargement and mask cost reduction,” manuscript submitted to *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 2012, under revision.

International Conference Papers

[1] **Y. Qu**, C. H. Heng, A. Tay and T. H. Lee, "OPC mask simplification using over-designed timing slack of standard cells " in *Proc. of the SPIE - Microtechnologies 2013*, Grenoble, France, p. 876425, 2013.

[2] **Y. Qu**, C. H. Heng, A. Tay and T. H. Lee, "Fast optical proximity correction with timing optimization ready standard cells" in *Proc. of the SPIE - Design for Manufacturability through Design-Process Integration VI*, San Jose, CA, p. 832714, 2012.

[3] Y. S. Ngo, **Y. Qu**, A. Tay and T. H. Lee, "In-situ critical dimension control during post-exposure bake with spectroscopic ellipsometry" in *Proc. of the SPIE - Metrology, Inspection, and Process Control for Microlithography XXVI*, San Jose, CA, p. 83242N, 2012.

[4] **Y. Qu**, A. Tay and T. H. Lee, "Iterative feedback tuning of optical proximity correction mask in lithography," in *Proc. of IEEE/SICE International Symposium on System Integration*, Kyoto, Japan, pp. 851-856, 2011.

[5] **Y. Qu**, A. Tay and T. H. Lee, "Feedback control applied to improve convergence of performance-based optical proximity correction in advanced lithography," in *Proc. of IASTED International Conference on Control and Applications*, Banff, Canada, pp. 593-599, 2010.

[6] **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. of IEEE/SEMI Advanced Semiconductor Manufacturing Conference (ASMC 2010)*, San Francisco, CA, pp. 99-103, 2010. (Best student paper award nomination)

Bibliography

[1] S. Landis, *Lithography: main techniques.* Wiley, Dec 2010.

[2] E. Muzio, "Lithography coo analysis," SEMATECH, Tech. Rep., 2000. [Online]. Available: http://www.sematech.org

[3] A. Wuest, A. Hazelton, and G. Hughes, "Estimation of cost comparison of lithography technologies at the 22 nm half-pitch node," in *Proc. SPIE - Alternative Lithographic Technologies*, San Jose, CA, Feb 2009, p. 72710Y.

[4] H. Khorram, K. Nakano, N. Sagawa, T. Fujiwara, Y. Iriuchijima, T. Sei, T. Takahiro, K. Nakamura, K. Shiraishi, and T. Hayashi, "Cost of ownership/yield enhancement of high volume immersion lithography using topcoat-less resists," *IEEE Trans. on Semiconductor Manufacturing*, vol. 24, no. 2, pp. 173–181, May 2011.

[5] B. J. Lin, *Optical Lithography: Here is Why.* SPIE Press, Feb 2010.

[6] S. Campbell, *Fabrication Engineering at the Micro and Nanoscale (3rd Ed).* Oxford, Sep 2007.

[7] C. Mack, *Fundamental Principles of Optical Lithography: the Science of Microfabrication.* Wiley, Dec 2007.

[8] G. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 25, no. 6, pp. 114–117, April 1965.

[9] ITRS, "International technology roadmap for semiconductors," SEMATECH, Tech. Rep., 2010. [Online]. Available: http://www.itrs.net

[10] X. Ma and G. Arce, *Computational lithography.* Wiley, Sep 2010.

[11] Mentor, *Calibre Litho-Friendly Design Users Manual*, 2011.

[12] A. Wong, *Resolution enhancement techniques in optical lighography.* SPIE Press, Mar 2001.

[13] C. Spence, "Full-chip lithography simulation and design analysis: how opc is changing ic design," in *Proc. of SPIE - Advances in resist technology and processing XXII*, San Jose, CA, Feb 2005, pp. xix–xxxii.

[14] P. Gupta, A. Kahng, D. Sylvester, and J. Yang, "Performance driven opc for mask cost reduction," *J. Micro/Nanolith. MEMS MOEMS*, vol. 6, no. 3, p. 031005, July 2007.

[15] N. Cobb, "Fast optical and process proximity correction algorithms for integrated circuit manufacturing," Ph.D. dissertation, Univ. California, Berkeley, 1998.

[16] H. Hopkins, "On the diffraction theory of optical images," *Proc. of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 217, no. 1130, pp. 408–432, May 1953.

[17] C. Spence, "Full-chip lithography simulation and design analysis: how opc is changing ic design," in *Proc. SPIE - Emerging Lithographic Technologies IX*, San Jose, CA, Feb 2005, pp. 1–14.

[18] Mentor, *Calibre Workbench User Manual*, 2011.

[19] N. Chung, Y. Yoon, S. Lee, S. Kim, S. Ha, and S. Lee, "The gate cd uniformity improvement by the layout retarget with refer to the litho process," in *Proc. SPIE - Optical Microlithography XX*, San Jose, CA, Feb 2007, p. 652042.

[20] Y. Trouiller, J. Serrand, C. Miramond, Y. Rody, S. Manakli, and P. Goirand, "Arf imaging with off-axis illumination and subresolution assist bars: a compromise between mask constraints and lithographic process constraints," in *Proc. SPIE - Optical Microlithography XV*, Santa Clara, CA, Mar 2002, p. 1522C1529.

[21] S. Banerjee, P. Elakkumanan, L. Liebmann, and M. Orshansky, "Electrically driven optical proximity correction based on linear programming," in *IEEE/ACM Int. Conf. Comput.-Aided Des.*, Piscataway, NJ, Nov 2008, pp. 473–479.

[22] S. Teh, C. Heng, and A. Tay, "Performance-based optical proximity

correction methodology," *IEEE Trans. on Comp.-Aided Des. of Int. Cir. and Sys.*, vol. 29, no. 1, pp. 51–64, Jan 2010.

[23] Y. Zhang, R. Gray, S. Chou, B. Rockwell, G. Xiao, H. Kamberian, R. Cottle, A. Wolleben, and C. Progler, "Mask cost analysis via write time estimation," in *Proc. SPIE - Design and Process Integration for Microelectronic Manufacturing III*, San Jose, CA, May 2005, pp. 313–318.

[24] H. Zhang, Y. Du, M. Wong, and K. Chao, "Mask cost reduction with circuit performance consideration for self-aligned double patterning," in *Proc. Asia and South Pacific Design Automation Conference*, Yokohama, Japan, Jan 2011, pp. 787–792.

[25] T. Jhaveri, V. Rovner, L. Liebmann, L. Pileggi, A. Strojwas, and J. Hibbeler, "Co-optimization of circuits, layout and lithography for predictive technology scaling beyond gratings," *IEEE Trans. on Comp.-Aided Des. of Int. Cir. and Sys.*, vol. 29, no. 4, pp. 509–527, Apr 2010.

[26] D. Reinhard and P. Gupta, "On comparing conventional and electrically driven opc techniques," in *Proc. SPIE - Photomask Technology*, Monterey, CA, Sep 2009, p. 748838.

[27] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital integrated circuits: a design perspective (2nd Ed)*. Prentice Hall, Jan 2003.

[28] C. Li, L. Milor, C. Ouyang, M. Maly, and Y. Peng, "Analysis of the

impact of proximity correction algorithms on circuit performance," *IEEE Trans. on Semiconductor Manufacturing*, vol. 12, no. 3, pp. 313–322, Aug 1999.

[29] M. Orshansky, L. Milor, C. Pinhong, K. Keutzer, and C. Hu, "Impact of spatial intrachip gate length variability on the performance of high-speed digital circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 5, pp. 544–553, May 2002.

[30] M. Orshansky, L. Milor, and C. Hu, "Characterization of spatial intrafield gate cd variability, its impact on circuit performance, and spatial masklevel correction," *IEEE Trans. on Semiconductor Manufacturing*, vol. 17, no. 1, pp. 2–11, Feb 2004.

[31] M. Orshansky and L. Milor, "Impact on circuit performance of deterministic within-die variation in nanoscale semiconductor manufacturing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 7, pp. 1350–1367, Jul 2006.

[32] K. Herold, N. Chen, and I. Stobert, "Managing high accuracy and fast convergence in opc," in *Proc. of SPIE - Photomask Technology*, Monterey, CA, Sep 2006, p. 634924.

[33] S. Choi, A. Je, J. Hong, M. Yoo, and J. Kong, "Meef-based correction to achieve opc convergence of low-k1 lithography with strong oai," in

Proc. of SPIE - Optical Microlithography XIX, San Jose, CA, Feb 2006, p. 61540P.

[34] P. Yu and D. Pan, “A novel intensity based optical proximity correction algorithm with speedup in lithography simulation,” in *Proc. of IEEE/ACM International Conference on Computer Aided Design*, San Jose, CA, Nov 2007, pp. 854–859.

[35] C. Wang, Q. Liu, and L. Zhang, “Accelerate opc convergence with new iteration control methodology,” in *Proc. of SPIE - Photomask Technology*, Monterey, CA, Oct 2008, p. 712240.

[36] W. Poppe, L. Capodieci, J. Wu, and A. Neureuther, “From poly line to transistor: Building bsim models for non-rectangular transistors,” in *Proc. SPIE - Design and Process Integration for Microelectronic Manufacturing IV*, San Jose, CA, Feb 2006, p. 61560P.

[37] B. Painter, L. Melvin III, and M. Rieger, “Classical control theory applied to opc correction segment convergence,” in *Proc. of SPIE - Optical Microlithography XVII*, Santa Clara, CA, May 2004, pp. 1198–1206.

[38] H. Hjalmarsson, S. Gunnarsson, and M. Gevers, “A convergent iterative restricted complexity control design scheme,” in *Proc. IEEE Conf. on Decision & Control*, Lake Buena Vista, FL, Dec 1994, pp. 1735–1740.

[39] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin, “Iterative

feedback tuning: theory and applications," *IEEE Control Systems Magazine*, vol. 18, no. 4, pp. 26–41, Aug 1998.

[40] A. Tay, W. Ho, J. Deng, and B. Lok, "Control of photoresist film thickness: Iterative feedback tuning approach," *Computers and Chemical Engineering*, vol. 30, no. 3, pp. 572–579, Jan 2006.

[41] Nangate, *45nm Open Cell Library*, 2009. [Online]. Available: http://www.nangate.com/openlibrary

[42] P. Gupta, F. Heng, and M. Lavin, "Merits of cellwise model-based opc," in *Proc. SPIE - Design and Process Integration for Microelectronic Manufacturing II*, Santa Clara, CA, Feb 2004, pp. 182–189.

[43] S. Sun, C. Bencher, Y. Chen, H. Dai, M. Cai, J. Jin, P. Blanco, L. Miao, P. Xu, X. Xu, J. Yu, R. Hung, S. Oemardani, O. Chan, C. Chang, and C. Ngai, "Demonstration of 32 nm half-pitch electrical testable nand flash patterns using self-aligned double patterning," in *Proc. of SPIE - Optical Microlithography XXII*, San Jose, CA, Feb 2009, p. 72740D.

[44] Y. Chen, P. Xu, L. Miao, Y. Chen, X. Xu, D. Mao, P. Blanco, C. Bencher, R. Hung, and C. Ngai, "Self-aligned triple patterning for continuous ic scaling to half-pitch 15nm," in *Proc. of SPIE - Optical Microlithography XXIV*, San Jose, CA, Feb 2011, p. 79731P.

[45] G. Franklin, J. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems (6th Ed)*. Prentice Hall, Oct 2009.

[46] Y. Su, P. Ng, K. Tsai, and Y. Chen, “Design of automatic controllers for model-based opc with optimal resist threshold determination for improving correction convergence,” in *Proc. of SPIE - Optical Microlithography XXI*, San Jose, CA, Mar 2008, p. 69243Z.

[47] Synopsys, *HSpice Application Manual*, 2010.

[48] PTM, *PTM HP/LP models*, 2008. [Online]. Available: http://ptm.asu.edu/latest.html

[49] S. Banerjee, K. Agarwal, C. Sze, S. Nassif, and M. Orshansky, “A methodology for propagating design tolerances to shape tolerances for use in manufacturing,” in *Proc. of the Design, Automation, and Test in Europe (DATE) Conference*, Dresden, Germany, Mar 2010, pp. 1273 – 1278.

[50] Y. Ban, S. Sundareswaran, and D. Pan, “Total sensitivity based dfm optimization of standard library cells,” in *Proc. of the International Symposium on Physical Design (ISPD)*, San Francisco, CA, Mar 2010, pp. 113–120.

[51] P. Yu, S. Shi, and D. Pan, “Process variation aware opc with variational lithography modeling,” in *Proc. Design Automation Conference*, San Francisco, CA, Jul 2006, pp. 785–790.

[52] J. Yang, L. Capodieci, and D. Sylvester, “Advanced timing analysis

based on post-opc extraction of critical dimensions," in *Proc. Design Automation Conference*, Anaheim, CA, Jun 2005, pp. 359–364.

[53] C. C. Pan, M. and H. Zhou, "Timing yield estimation using statistical static timing analysis," in *Proc. IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005, pp. 2461–2464.

[54] E. Yang, C. Li, X. Kang, and E. Guo, "Model-based retarget for 45 nm node and beyond," in *Proc. SPIE - Optical Microlithography XXII*, San Jose, CA, Feb 2009, p. 727428.

[55] S. Banerjee, K. Agarwal, and M. Orshansky, "Smato: Simultaneous mask and target optimization for improving lithographic process window," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, Nov 2010, pp. 100–6.

[56] K. Agarwal and S. Banerjee, "Integrated model-based retargeting and optical proximity correction," in *Proc. SPIE - Design and Process Integration for Microelectronic Manufacturing V*, San Jose, CA, Feb 2011, p. 79740F.

[57] H. Levinson, M. McCord, R. Cerrina F., Allen, J. Skinner, A. Neureuther, M. Peckerar, F. Perkins, M. Rooks, and P. Rai-Choudhury, *Handbook of Microlithography, Micromachining, and Microfabrication. Volume 1: Microlithography.* SPIE Press, March 1997.

[58] Synopsys, *Design Compiler User Guide*, 2010.

[59] W. Elmore, “The transient response of damped linear networks with particular regard to wideband amplifiers,” *J. Applied Physics*, vol. 19, no. 1, Jan 1948.

[60] J. Torres and C. Berglund, “Integrated circuit dfm framework for deep sub-wavelength processes,” in *Proc. SPIE - Design and Process Integration for Microelectronic Manufacturing III*, San Jose, CA, May 2005, p. 575639.

[61] T. Chan, A. Kagalwalla, and P. Gupta, “Measurement and optimization of electrical process window,” *J. Micro/Nanolith. MEMS MOEMS*, vol. 10, no. 1, p. 013014, Jan 2011.

[62] R. Fathy, M. Al-Imam, H. Diab, M. Fakhry, J. Torres, B. Graupp, J. Brunet, and M. Bahnas, “Litho aware method for circuit timing/power analysis through process,” in *Proc. SPIE - Design and Process Integration for Microelectronic Manufacturing V*, no. Feb, San Jose, CA, 2007, p. 65210O.

[63] E. Chin, “Compensation for lithography induced process variations during physical design,” Ph.D. dissertation, Univ. California, Berkeley, 2011.

[64] H. Zhang, Y. Du, M. Wong, and K. Chao, “Characterization of the performance variation for regular standard cell with process

nonidealities," in *Proc. SPIE - Design and Process Integration for Microelectronic Manufacturing V*, San Jose, CA, Feb 2011, p. 79740T.

[65] ISCAS, *ISCAS High-Level Models*, 1985. [Online]. Available: http://www.eecs.umich.edu/ jhayes/iscas/

[66] Cadence, *Encounter User Guide*, 2011.

[67] J. Qi, "Study on preparation and validation of nanometer-scale opc," Master's thesis, Zhejiang University, China, 2012.

[68] A. Sreedhar and S. Kundu, "Statistical timing analysis based on simulation of lithographic process," in *Proc. IEEE International Conference on Computer Design*, Lake Tahoe, CA, Oct 2009, pp. 29–34.

[69] L. Pang and B. Nikolic, "Measurements and analysis of process variability in 90nm cmos," *IEEE Journal of Solid-state Circuits*, vol. 44, no. 5, pp. 1655–1663, May 2009.

[70] K. Yelamarthi and C. Chen, "Timing optimization and noise tolerance for dynamic cmos susceptible to process variations," *IEEE Trans. on Semiconductor Manufacturing*, vol. 25, no. 2, pp. 255–265, May 2012.

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

[72] A. Kahng, S. Muddu, and C. Park, “Auxiliary pattern-based opc for better printability, timing and leakage control,” *J. Micro/Nanolith. MEMS MOEMS*, vol. 7, no. 1, p. 013002, Jan 2008.

[73] X. Yan, Z. Shi, Y. Chen, and Q. Chen, “Advances in opc technology and development of zopc tool,” in *Proc. SPIE - Quantum Optics, Optical Data Storage, and Advanced Microlithography*, Beijing, China, Nov 2007, p. 68271U.

[74] S. Teh, C. Heng, and A. Tay, “Library-based performance-based opc,” in *Proc. SPIE - Design and Process Integration for Microelectronic Manufacturing IV*, San Jose, CA, Feb 2010, p. 76410X.

[75] C. Fang and W. Jone, “Timing optimization by gate resizing and critical path identification,” *IEEE Trans. on Comp.-Aided Des. of Int. Cir. and Sys.*, vol. 14, no. 2, pp. 201–216, Feb 1995.

[76] Synopsys, *PrimeTime User Guide*, 2010.

[77] G. E. Bailey, A. Tritchkov, J.-W. Park, L. Hong, V. Wiaux, E. Hendrickx, S. Verhaegen, P. Xie, and J. Versluijs, “Double pattern eda solutions for 32nm hp and beyond,” in *Proc. SPIE - Design for Manufacturability through Design-Process Integration V*, no. Feb, San Jose, CA, 2007, p. 65211K.

[78] K. Lucas, C. Cork, A. Miloslavsky, G. Luk-Pat, L. Barnes, J. Hapli, J. Lewellen, G. Rollins, V. Wiaux, and S. Verhaegen, “Double-patterning

interactions with wafer processing, optical proximity correction, and physical design flows," *J. Micro/Nanolith. MEMS MOEMS*, vol. 8, no. 3, p. 033002, July 2009.

[79] C. Mack, *Field Guide to Optical Lithography.* SPIE Press, Jan 2006.

[80] C. Clifford, Y. Zou, A. Latypov, O. Kritsun, T. Wallow, H. Levinson, F. Jiang, D. Civay, K. Standiford, R. Schlief, L. Sun, O. Wood, S. Raghunathan, P. Mangat, H. Koh, C. Higgins, J. Schefske, and M. Singh, "Euv opc for the 20-nm node and beyond," in *Proc. of SPIE - Extreme Ultraviolet (EUV) Lithography III*, San Jose, CA, Feb 2012, p. 83221M.

[81] Y. Wei and B. Robert, *Advanced Processes for 193-nm Immersion Lithography.* SPIE Press, Feb 2009.

[82] A. Tay, W. Ho, and Y. Poh, "Minimum time control of conductive heating systems for microelectronics processing," *IEEE Trans. on Semiconductor Manufacturing*, vol. 14, no. 4, pp. 381–386, Nov 2001.

[83] A. Tay, W. Ho, C. Schaper, and L. Lee, "Constraint feedforward control for thermal processing of quartz photomasks in microelectronics manufacturing," *Journal of Process Control*, vol. 14, no. 1, pp. 31–39, Feb 2004.

[84] A. Tay, W. Ho, X. Wu, and X. Chen, "In situ monitoring of photoresist thickness uniformity of a rotating wafer in lithography," *IEEE Trans. on*

Instrumentation and Measurement, vol. 58, no. 12, pp. 3978–3984, Dec 2009.

Appendix A

Iterative Feedback Tuning Derivation

In this appendix, only the equations necessary for implementing the OPC mask are reviewed. Detailed discussion of the algorithms can be found in [38–40].

A PI controller, $C(\rho)$, can be described as follows:

$$C(\rho) = K_P + \frac{K_I}{q-1}, \tag{A.1}$$

where K_P is the proportional gain, K_I is the integral gain, q is the time shift operator and $\rho = \begin{bmatrix} K_P & K_I \end{bmatrix}^T$. The predicted circuit performance and the control signal with controller parameter ρ is denoted as $y(\rho)$ and $u(\rho)$ respectively. The difference between $y(\rho)$ and reference signal r (desired circuit performance) is:

$$\widetilde{y}(\rho) = y(\rho) - r \tag{A.2}$$

The target of IFT is to minimize a quadratic criterion:

$$J(\rho) = \frac{1}{2N}\left[\sum_{t=1}^{N}\left(L_y\widetilde{y}_t(\rho)\right)^2 + \eta\sum_{t=1}^{N}\left(L_u u_t(\rho)\right)^2\right], \tag{A.3}$$

where N is the number of data points, $\widetilde{y}_t(\rho)$ and $u_t(\rho)$ denotes the sampled values of $\widetilde{y}(\rho)$ and $u(\rho)$ at time t. L_y and L_u are the weights to set penalty on the two terms in (A.3) which are simply set to 1 in this work. The value of ρ which minimizes this quadratic criterion is equal to:

$$\rho^* = \arg \underset{\rho}{min} J(\rho), \tag{A.4}$$

where ρ^* is actually a solution to the equation:

$$\frac{\partial J(\rho)}{\partial \rho} = 0 = \frac{1}{N}\left[\sum_{t=1}^{N} \widetilde{y}_t(\rho)\frac{\partial \widetilde{y}_t(\rho)}{\partial \rho} + \eta \sum_{t=1}^{N} u_t(\rho)\frac{\partial u_t(\rho)}{\partial \rho}\right], \tag{A.5}$$

and ρ^* can be obtained iteratively by a Newton-like algorithm:

$$\rho_{i+1} = \rho_i - \gamma R_i^{-1}\frac{\partial J(\rho)}{\partial \rho}, \tag{A.6}$$

where i is the iterative feedback tuning iteration number, γ is a positive real scalar to determine the step size, and R_i is an appropriate positive definite matrix, typically an estimation of the Hessian of $J(\rho)$. A Gauss-Newton approximation of the Hessian can be used:

$$R_i = \frac{1}{N}\left[\sum_{t=1}^{N} \frac{\partial \widetilde{y}_t(\rho)}{\partial \rho}\frac{\partial \widetilde{y}_t(\rho)}{\partial \rho}^T + \eta \sum_{t=1}^{N} \frac{\partial u_t(\rho)}{\partial \rho}\frac{\partial u_t(\rho)}{\partial \rho}^T\right] \tag{A.7}$$

The values of $\widetilde{y}_t(\rho)$ and $u_t(\rho)$ can be recorded, but $\dfrac{\partial \widetilde{y}_t(\rho)}{\partial \rho}$ and $\dfrac{\partial u_t(\rho)}{\partial \rho}$ cannot be measured directly. The followings gives an approach to derive them. For the closed loop system with plant G in Figure 2.5, we have:

$$y(\rho) = \frac{C(\rho)G}{1 + C(\rho)G}r \tag{A.8}$$

$$u(\rho) = C(\rho)(r - y(\rho)) \tag{A.9}$$

Differentiating $\widetilde{y}(\rho)$ and $u(\rho)$ with respect to ρ, we have:

$$\frac{\partial \widetilde{y}(\rho)}{\partial \rho} = \frac{\partial y(\rho)}{\partial \rho} = \frac{1}{C(\rho)} \frac{\partial C(\rho)}{\partial \rho} \left[\frac{C(\rho)G}{1+C(\rho)G}(r - y(\rho)) \right] \tag{A.10}$$

$$\frac{\partial u(\rho)}{\partial \rho} = \frac{1}{C(\rho)} \frac{\partial C(\rho)}{\partial \rho} \left[\frac{C(\rho)}{1+C(\rho)G}(r - y(\rho)) \right] \tag{A.11}$$

The above mentioned relationships suggest to conduct a pair of simulation runs, so that $\frac{\partial \widetilde{y}_t(\rho)}{\partial \rho}$ and $\frac{\partial u_t(\rho)}{\partial \rho}$ can be measured:

- Run 1: with controller parameters ρ, run the whole system with normal reference input r, and record the output y and control signal u;
- Run 2: with controller parameters ρ, change input to $r - y$ (r and y are recorded in Run 1), record the output y^e and control signal u^e in this run (the superscript e means the difference between r and y).

In Run 2, we have the followings:

$$y^e(\rho) = \frac{C(\rho)G}{1+C(\rho)G}(r - y(\rho)) \tag{A.12}$$

$$u^e(\rho) = C(\rho)\left[(r - y(\rho)) - y^e(\rho)\right] = \frac{C(\rho)}{1+C(\rho)G}(r - y(\rho)) \tag{A.13}$$

Therefore, (A.10) and (A.11) becomes:

$$\frac{\partial \widetilde{y}(\rho)}{\partial \rho} = \frac{\partial y(\rho)}{\partial \rho} = \frac{1}{C(\rho)} \frac{\partial C(\rho)}{\partial \rho} y^e(\rho) \tag{A.14}$$

$$\frac{\partial u(\rho)}{\partial \rho} = \frac{1}{C(\rho)} \frac{\partial C(\rho)}{\partial \rho} u^e(\rho) \tag{A.15}$$

For PI controller in the form of (A.1), we can further derive:

$$\begin{bmatrix} \frac{\partial \widetilde{y}(\rho)}{\partial K_P} \\ \frac{\partial \widetilde{y}(\rho)}{\partial K_I} \end{bmatrix} = \begin{bmatrix} \frac{1}{K_P} y^e(\rho) \\ \frac{1}{K_P(q-1)+K_I} y^e(\rho) \end{bmatrix} \tag{A.16}$$

$$\begin{bmatrix} \dfrac{\partial \widetilde{u}(\rho)}{\partial K_P} \\ \dfrac{\partial \widetilde{u}(\rho)}{\partial K_I} \end{bmatrix} = \begin{bmatrix} \frac{1}{K_P} u^e(\rho) \\ \\ \frac{1}{K_P(q-1)+K_I} u^e(\rho) \end{bmatrix} \tag{A.17}$$

To sum up, by conducting a pair of simulation runs, (A.17), (A.16), (A.5), (A.7), and (A.6) can be used to calculate the new controller parameter, *i.e.* ρ_{i+1}. By repeating this, the minimum $J(\rho)$ and the solution ρ^* can be gradually approached. ρ^* can be chosen as the optimal controller parameters for the PI controller in (A.1).

Appendix B

Timing Process Window of PWA-OPC

From Zhangs model [64], the geometry process window (GPW) can be defined as:

$$GPW = \left\{ (f,d) \,|L_0 - \Delta L \leq L \leq L_0 + \Delta L, f \in F, d \in D \right\}, \tag{A.18}$$

where (f,d) are the points in focus-dosage domain, f refers to defocus and d refers to dosage, L_0 is the nominal gate length, ΔL is the tolerance range (typically 10% of L_0), F and D refer to the defocus and dosage value bounds, *i.e.* process variation ranges. Typical plots of GPW can be found in Ref. [64].

In digital circuit, there are usually two extreme values to define the bond of timing delay: best case timing T_{BC} and worst case timing T_{WC}. They are both functions of gate length L:

$$T_{BC} = \mathcal{F}_{BC}(L),$$

$$T_{WC} = \mathcal{F}_{WC}(L),$$

where $\mathcal{F}_{BC}$ and $\mathcal{F}_{WC}$ can be modeled using methods in [27]. Therefore, the timing process window is defined as:

$$TPW = \left\{(f,d)\,|T_0 - \Delta T \le T_{BC} \le T_0 + \Delta T, T_0 - \Delta T \le T_{WC} \le T_0 + \Delta T, f \in F, d \in D\right\}, \tag{A.19}$$

where T_0 is the nominal timing and ΔT is the tolerance range in timing domain. TPW can also be expressed as a the intersection of two sub process windows:

$$TPW = TPW_{BC} \cap TPW_{WC}, \tag{A.20}$$

where

$$TPW_{BC} = \left\{(f,d)\,|T_0 - \Delta T \le T_{BC} \le T_0 + \Delta T, f \in F, d \in D\right\},$$
$$TPW_{WC} = \left\{(f,d)\,|T_0 - \Delta T \le T_{WC} \le T_0 + \Delta T, f \in F, d \in D\right\},$$

For ideal lithography systems, where printed images are identical to designed shapes, the values of T_{BC} and T_{WC} are equal. The area of TPW is of maximum size. However, as lithography always distorts printed images, the timing delay varies accordingly. For good process-window-aware methods, the variation range of timing delay is smaller than non-process-window-aware methods, *i.e.*:

$$|T_{BC,PWA} - T_{WC,PWA}| < |T_{BC,nonPWA} - T_{WC,nonPWA}|\,. \tag{A.21}$$

The above equation indicates that the areas of TPW of PWA methods are larger than non-PWA methods, since the intersection of TPW_{BC} and TPW_{WC} of PWA methods is larger.