The Dissertation Committee for Jhih-Rong Gao
certifies that this is the approved version of the following dissertation:

Lithography Aware Physical Design and Layout Optimization for Manufacturability

Committee:

David Z. Pan, Supervisor

Bikram Baidya

Andreas Gerstlauer

Michael Orshansky

Nur Touba

Lithography Aware Physical Design and Layout Optimization for Manufacturability

by

Jhih-Rong Gao, B.S.; M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2014

Dedicated to my parents.

Acknowledgments

Everything starts from April 1, 2009, when Professor David Z. Pan offered me to work in his research group. I am truly lucky and sincerely grateful to have Dr. Pan as my Ph.D. supervisor. Thank you for seeing my potential, giving me guidance when I was stuck, and encouraging me to pursuit greater achievements. Your passion for research sets the model of a great scholar that I would love to follow. This dissertation would not have been possible without your help and support.

I would like to express my appreciation to my committee members, Dr. Bikram Baidya, Dr. Andreas Gerstlauer, Dr. Michael Orshansky, and Dr. Nur Touba. Thank you for kindly providing advice and pointing out ways to improve my works. Special thanks to Dr. Baidya who offered me great insights and help for my internship and dissertation. I would like to thank Dr. Prashant Saxena for being a great mentor during my internship at Synopsys. My sincere thanks also go to Atul S. Walimbe and Dr. Prasad N. Atkar for your active participation and guidance in my internship at Intel.

I am grateful to have worked with my fellow UTDA members, Duo Ding, James Ban, Kun Yuan, Ashutosh Chakraborty, Bei Yu, Xiaoqing Xu, Jiwoo Pak, and Yilin Zhang. Only through those helpful discussion and harsh critiques that I could keep polishing my research.

I dedicate my dissertation to my parents Yu-Ming and Jung-Hua. No word can express my gratitude to Yu-Lin for bringing me to the world. Without your unlimited love and support, I could not pursue and accomplish my goals. Thanks to my brother, Chung-Wei, who takes good care of the family when I could not be there. I owe my sincere thanks to my family members Shi-Ying and DingChuen. Life would be less sweet without your company. To all my friends at Austin, it is my pleasure to have met you here. Thanks everyone who has participated this long but fruitful journey.

Lithography Aware Physical Design and Layout Optimization for Manufacturability

Publication No. _____

Jhih-Rong Gao, Ph.D.
The University of Texas at Austin, 2014

Supervisor: David Z. Pan

As technology continues to scale down, semiconductor manufacturing with 193nm lithography is greatly challenging because the required half pitch size is beyond the resolution limit. In order to bridge the gap between design requirements and manufacturing limitations, various resolution enhancement techniques have been proposed to avoid potentially problematic patterns and to improve product yield. In addition, co-optimization between design performance and manufacturability can further provide flexible and significant yield improvement, and it has become necessary for advanced technology nodes. This dissertation presents the methodologies to consider the lithography impact in different design stages to improve layout manufacturability.

Double Patterning Lithography (DPL) has been a promising solution for sub-22nm node volume production. Among DPL techniques, self-aligned double patterning (SADP) provides good overlay controllability when two

masks are not aligned perfectly. However, SADP process places several limitations on design flexibility and still exists many challenges in physical design stages. Starting from the early design stage, we analyze the standard cell designs and construct a set of SADP-aware cell placement candidates, and show that placement legalization based on this SADP awareness information can effectively resolve DPL conflicts. In the detailed routing stage, we propose a new routing cost formulation based on SADP-compliant routing guidelines, and achieve routing and layout decomposition simultaneously. In the case that limited routing perturbation is allowed, we propose a post-routing flow based on lithography simulation and lithography-aware design rules. Both routing methods, one in detailed routing stage and one in post routing stage, reduce DPL conflicts/violations significantly with negligible wire length impact. In the layout decomposition stage, layout modification is restricted and thus the manufacturability is even harder to guaranteed. By taking the advantage of complementary lithography, we present a new layout decomposition approach with e-beam cutting, which optimizes SADP overlay error and e-beam lithography throughput simultaneously.

After the mask layout is defined, optical proximity correction (OPC) is one of the resolution enhancement techniques that is commonly required to compensate the image distortion from the lithography process. We propose an inverse lithography technique to solve the OPC problem considering design target and process window co-optimization. Our mask optimization is pixel based and thus can enable better contour fidelity. In the final physi-

cal verification stage, a complex and time-consuming lithography simulation needs to be performed to identify faulty patterns. We provide a classification method based on support vector machine and principle component analysis that detects lithographic hotspots efficiently and accurately.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Due to the delay in the next generation lithography technology such as Extreme Ultra Violate (EUV) [28] and E-beam lithography (EBL) [63, 82], the manufacturing industry still relies on the 193nm wavelength light source as illustrated in Fig. 1.1. As technology continues to scale down to sub-22nm, semiconductor manufacturing with 193nm lithography becomes greatly challenging because the required half pitch size is beyond the resolution limit. In order to bridge the gap between design requirements and manufacturing limi-

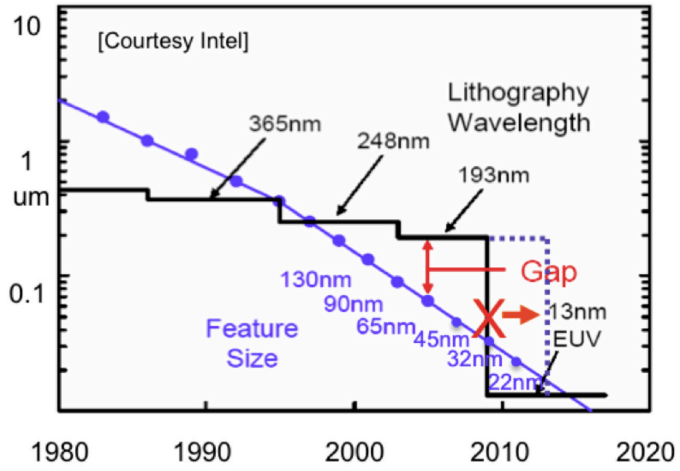


Figure 1.1: Optical lithography status [64]. Current lithography process still relies on the 193nm wavelength light source.

tations, various resolution enhancement techniques [64,80] have been proposed to improve product yield and avoid potentially problematic patterns. In addition, co-optimization between design performance and manufacturability can further provide flexible and significant yield improvement, and it has become necessary for advanced technology nodes.

1.1 Sub-wavelength Lithography Challenges

Double Patterning Lithography (DPL) has been a promising solution to achieve sub-22nm node volume production. In DPL, adjacent patterns with a distance less than the manufacturing limit must be decomposed and assigned to different masks. The decomposition process is referred to as coloring. Since each mask contains sparser patterns with larger spacing, it can be manufactured with the current optical lithography process. By combining patterns in the two masks together, the layout density is doubled and thus the lithography resolution is improved.

There are two main DPL schemes in current IC manufacturing: litho-etch-litho-etch (LELE) double patterning, and self-aligned double patterning (SADP). LELE [8,13,45,78,89] consists of two exposure and two etch processes as shown in Fig. 1.2 (a), and it allows stitch insertion to improve the decomposition flexibility. Applying stitches allows a pattern to be split into two masks; however, it makes patterns more sensitive to process variations. Besides, the alignment and magnification errors on the second mask exposure cause LELE to induce significant pattern overlay error [23] and thus degrade yield rate.

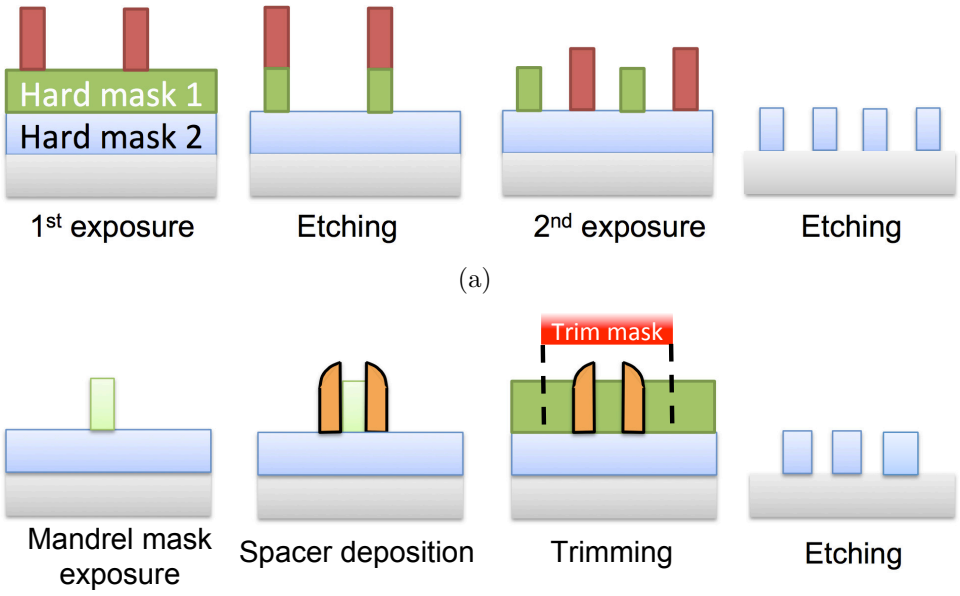
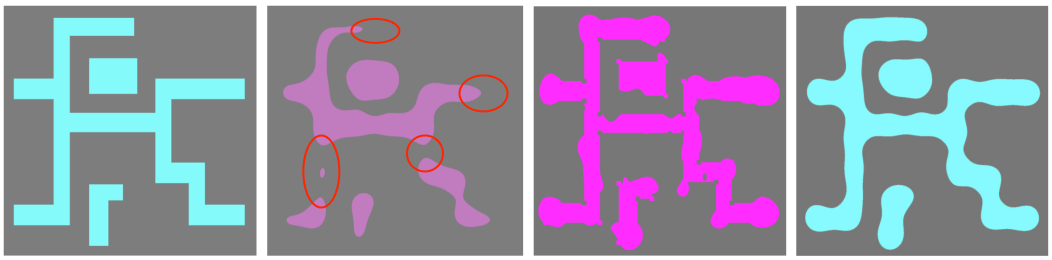


Figure 1.2: Manufacturing processes of the two main double pattering lithography techniques: (a) LELE and (b) SADP.

SADP, on the other hand, provides better overlay controllability than LELE DPL with its sidewall spacer based manufacturing process as shown in Fig. 1.2 (b). This has made SADP widely adopted for advanced technology nodes. However, SADP process places several limitations on design flexibility and still exists many challenges in physical design stages. Therefore, it is necessary to consider SADP compliance in early physical design stages to ensure SADP manufacturability.

In addition to the progress in manufacturing process such as DPL, various resolution enhancement techniques (RETs) have been proposed to achieve deep sub-wavelength lithography. Optical Proximity Correction (OPC) is one



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

Figure 1.3: Layout examples. (a) Target layout. (b) Printed image and lithography hotspots indicated by red circles. (c) OPC mask. (d) Improved printed image with OPC.

of the RETs that has been widely adopted. Fig. 1.3 (a) and (b) show a layout example and its corresponding printed image. Several open connections and line-end shortenings are caused by lithography effects, which could result in serious yield loss. With OPC technique, the mask is modified as shown in Fig. 1.3 (c) to compensate the image distortion, which generates better feature shapes in (d).

The area indicated by red circles in the above figure is also referred to as lithography hotspots and they should be eliminated in the physical design stages. With various design for manufacturability (DFM) techniques, at the last design stage, a physical verification still needs to be performed to identify hotspots and ensure design manufacturability. The hotspot detection problem is to locate hotspots on a given layout. Although conventional lithography simulation [47, 67] can accurately identify faulty layout patterns with complicated lithography models, the computational time is extremely expensive. Recently, hotspot detection methods based on machine learning and pattern-

ing matching have become popular candidates. Pattern matching techniques enable high accuracy and data learning algorithms provide high flexibility to adapt to new lithography processes and rules. However, how to achieve high detecting accuracy with low false alarms is still challenging.

1.2 Overview of this Dissertation

In this dissertation we present systematic methodologies [29–34] that consider the lithography impact in different design stages to improve layout manufacturability. Fig. 1.4 shows the typical design flow and our proposed lithography aware methodologies in the corresponding design stages.

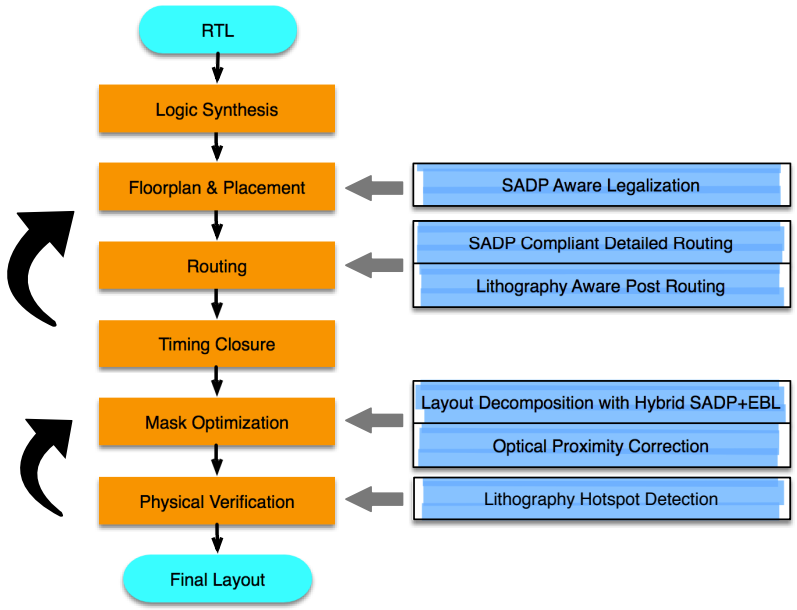


Figure 1.4: Summary of the design flow and the proposed methodologies in their corresponding design stages.

Our optimization for SADP friendliness is presented in Chapter 2 for different physical design stages, including (1) standard cell placement, where the arrangement of standard cells is decided considering the layout decomposability between cells; (2) detailed routing, where the pin-to-pin routes are determined considering the lithography effect between adjacent nets; and (3) post routing, where limited modification of routed paths is performed to reduce lithography hotspots; and (4) layout decomposition with complimentary e-beam lithography, where the hybrid pattern assignment among SADP masks and e-beams are determined.

In the current physical design flow, designs in RTL codes go through several procedures to generate the target layout. Layout decomposition for DPL is performed at a late stage before manufacturing. However, layout decomposability usually cannot be guaranteed especially for dense metal layers, and thus should be addressed in early design stages. We analyze the lithography impact on standard cells and find the arrangements that lead to decomposable layouts. Based on the eligible cell placement candidates, a SADP-aware legalization approach is presented in Section 2.2 to resolve coloring conflicts in standard cell level.

In addition to standard cell layers, lower routing layers are usually congested as well, which cause difficulty for layout decomposition. Integrating SADP awareness in routing stage provides a great flexibility to improve both decomposability and pattern overlay error. We propose a detailed routing approach in Section 2.3 to achieve routing and layout decomposition simulta-

neously based on SADP-friendly routing guidelines. Although this approach provides a full routing and DPL optimization, some designs may prefer to treat DFM guidelines as recommended rules rather than strictly forced DRC rules. Besides, in the late design stage, it is commonly demanded to fix lithographic violations without drastically changing existing routes. Based on these motivations, we present a SADP-compliant post routing flow to fix lithographic violations in Section 2.4.

Recently, complementary lithography that allows different lithography techniques work together has become promising. Since different lithography techniques have their own advantages and limitations, it is important to achieve good trade-off among these techniques. Although there has been related research on 1D layout, the co-optimization for hybrid lithography techniques on general 2D layouts is still challenging and under development. To improve the manufacturability of SADP-based designs, we propose a hybrid SADP and EBL layout decomposition approach in Section 2.5.

By applying the above physical design processes for SADP compliance, the layout can be optimized to ensure SADP manufacturability. To further improve the pattern quality during lithography process, OPC is a necessary step. The main objective for OPC is to obtain an optimized mask that can compensate the pattern distortion. As the feature size is getting smaller, the yield impact of layout uncertainty during the manufacturing process is getting larger. In Chapter 3, we propose a new OPC approach considering the design target optimization and process window minimization simultaneously.

Once the mask optimization is done, it is needed to verify if there are still lithography hotspots before delivering the design for manufacturing. In Chapter 4, we present a high performance hotspot detection approach based on principle component analysis and support vector machine. Several techniques, including hierarchical data clustering, data balancing, and multi-level training, are provided to enhance performance of the proposed approach. Our data clustering and data compression techniques help to improve the accuracy and reduce the false alarms.

Chapter 2

Self-aligned Double Patterning (SADP) Aware Physical Design

Self-aligned double patterning enables higher pitch resolution. However, SADP process requires valid layout decomposition to ensure its manufacturability which is difficult to guarantee especially for dense metal layers. Therefore, it is necessary to consider SADP compliancy in early physical design stages to ensure SADP manufacturability.

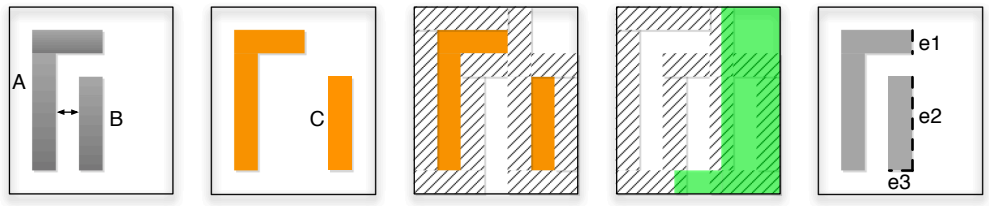
In this chapter, we first give an overview of SADP process and discuss its challenges in Section 2.1. A standard cell legalization approach for SADP decomposability is presented in Section 2.2. In Section 2.3, we propose a SADP-compliant detailed routing method to avoid conflicts among nets. To reduce the router burden, in Section 2.4, we present a post routing methodology that can identify and fix lithography unfriendly patterns efficiently. A layout decomposition with complimentary e-beam lithography is proposed in Section 2.5 that can provide higher design flexibility.

2.1 SADP Process Overview

In SADP, the double patterning spacing S_{dp} restricts the minimum spacing between two patterns on the same mask. Any two patterns with distance less than S_{dp} must not be fabricated on the same masks, otherwise, it is called a *conflict*. In general, the layout decomposition process involves decompose layout patterns into two sets; one is defined by the core (mandrel) mask, and the other is co-defined by spacers and the trim mask.

Fig. 2.1 shows an example to fabricate the target layout with SADP, where the arrow indicates a conflict between the two target patterns, meaning they cannot be fabricated on the same mask. Part of the target layout is first defined by the mandrel mask as shown in Fig. 2.1(b). Pattern C is called an *assist mandrel*, which helps to define target patterns but will not appear on the final layout. Next, a spacer material is deposited around the boundary of the mandrels as shown in the slashed area in (c). The mandrels will then be removed as shown in (d). After that, the second mask, trim mask shown in the green area will be applied to block the undesired layout region. A metal filling process will then fill the white area so that the final layout in (e) is obtained. We call pattern A a *mandrel pattern* since it is defined by the mandrel mask, and pattern B a *non-mandrel pattern*. To achieve a valid layout decomposition, all patterns on the mandrel and the trim mask must satisfy the minimum spacing S_{dp}.

One issue with SADP process is that the trim mask may not perfectly aligned to the mandrel mask. The resulted overlay error may cause line-end



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

Figure 2.1: SADP example. (a) Target layout patterns. (b) The mandrel mask. (c) Spacer deposition. (d) Mandrel removal and the trim mask. (e) Final patterns.

or CD variation; for example, a possible line-end shortening is on edge e_3. Spacers can work as an isolating material, and thus patterns that aligned to spacers would not suffer from overlay problem. Fig. 2.2 (a) and (b) show examples with and without overlay error, respectively. Pattern A is formed by the first exposure, while pattern B is formed by aligning to the boundaries of the spacer and the trim pattern. If the trim mask shifts, the right edge of pattern B would be vulnerable to the overlay error as shown in Fig. 2.2 (b), causing CD variation. A good layout decomposition should avoid patterns that are not aligned to spacers.

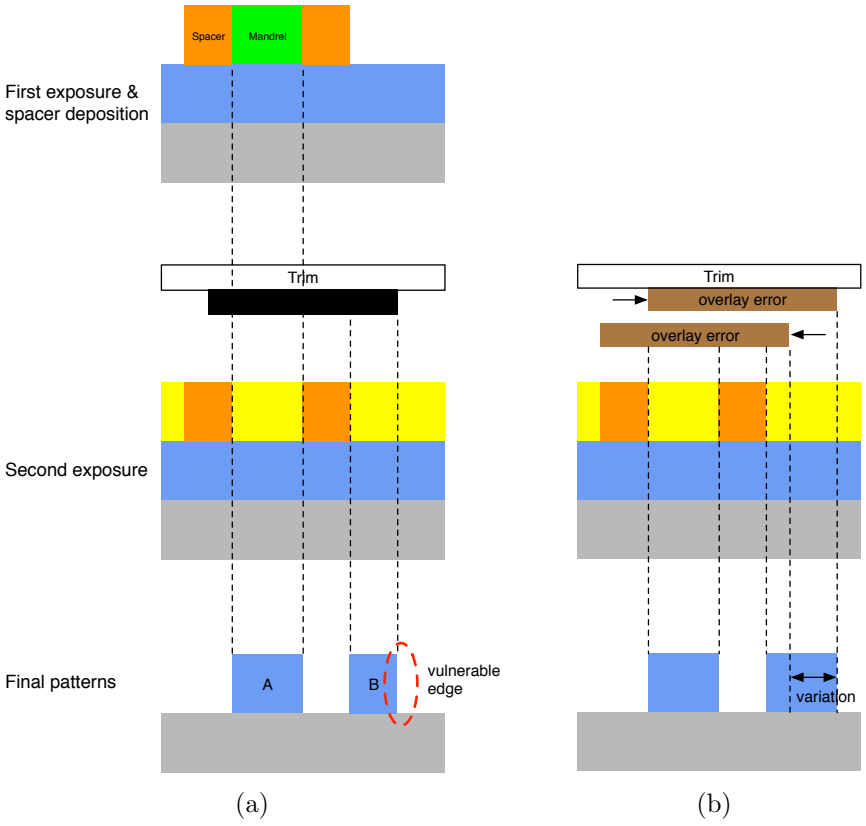


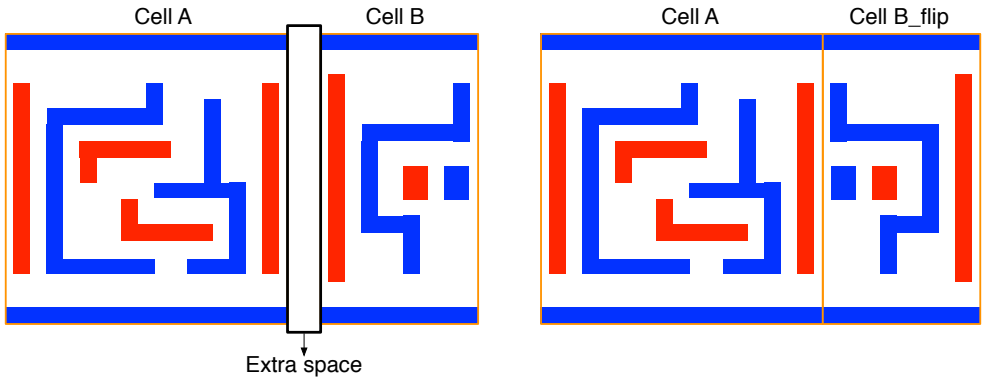
Figure 2.2: CD variation caused by SADP overlay error.

2.2 SADP Friendly Configuration for Standard Cell Library

The most dense metal layers in VLSI designs are metal1 and metal2 layers, which are commonly used by standard cells. Since the positions of standard cells are fixed after placement, potential coloring conflicts between adjacent cells needs to be detected and avoided to ensure the manufacturability for DPL.

Industries have been discussed the demand to integrate layout decomposition information into the placement stage. Previous studies [41, 52] focused on generating double patterning friendly standard cell library that helps to achieve layout decomposability after placement. They suggest restrictive design rules for standard cell designs to ensure decomposability, such as preassign color for power/ground net, force single color on each cell boundary, and remove the color dependency between power/ground net and signal nets. Based on a given standard cell library, Wassala et al. [70] proposed an approach to find all possible combinations of cell decompositions. However, the techniques in these previous works involve pattern splitting which cannot apply to SADP-based designs. Moreover, none of these works have evaluated the real impact of applying double patterning friendly standard cells in placement.

In current standard cell design methodology, designs are constructed with standard cells placed side by side in placement stage. Usually, layout decomposition is performed after placement and route. However, there may be potential conflicts caused by internal features of standard cells. The conflicts



(a) (b)

Figure 2.3: Standard cell placement examples. (a) Improper placement requires extra space to solve the coloring conflict in the boundary. (b) SADP compliant placement is obtained by flipping Cell B.

need to be solved by altering cell placement which may result in increase of the layout area. By brining double patterning aware information to the placement stage, it is possible to reduce the layout area and ease the burden for fixing conflicts in later stages [33, 81]. Fig. 2.3 shows two examples of placing two standard cells, where (a) shows an improper cell placement that generates a coloring conflict by the two red patterns on the cell boundary and needs extra space between cells to resolve the conflict; while (b) shows a more compact and conflict-free placement by simply flipping Cell B.

The rest of Section 2.2 is organized as follows. We will discuss the impact of standard cell placement in Section 2.2.1. Our SASP-aware legalization will be explained in Section 2.2.2. We will discuss our experimental results in Section 2.2.3, followed by the summary in Section 2.2.4. The preliminary results of this work were reported at [33].

2.2.1 Impact of Standard Cell Placement

2.2.1.1 The Turnaround to Placement Stage

Double pattering lithography enables further feature shrinking to sub-22nm technology. However, there exists a gap between the product of the design flow and a manufacturable layout. SADP process requires two adjacent patterns not fabricated on the same mask if the distance between them is less than the lithography resolution S_{dp}. Conflicts occur if the layout decomposition step fails to obey the spacing rule S_{dp} for all patterns on the same mask. Since the general physical design flow does not take SADP awareness into consideration, it is obvious that the layout after physical design flow may not be decomposable to enable SADP. Therefore, fixing loops as shown in Fig. 2.4 have to be iterated until the layout is ready for manufacturing.

Minor layout conflicts can possibly be fixed by layout decomposition

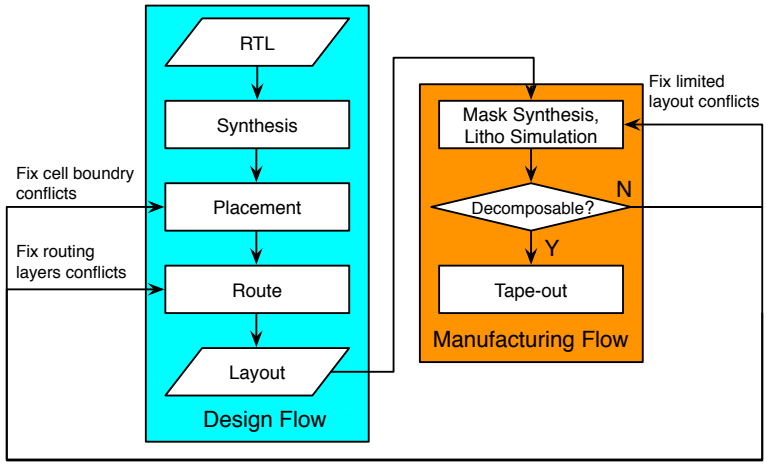


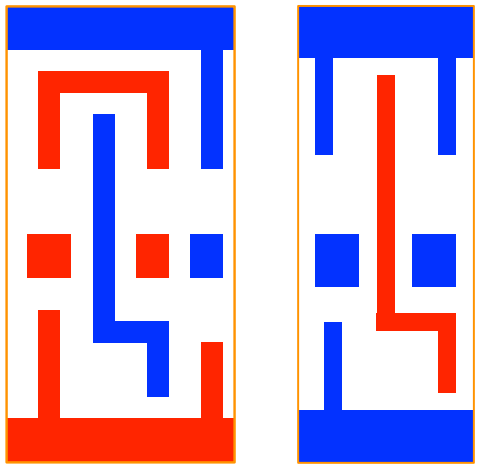
Figure 2.4: Design and manufacturing flow to enable deomposability.

techniques in mask synthesis stage. For example, SADP allows merging two conflicting patterns followed by line-cutting [51, 62] to generate the target layout. However, these kind of merging may be limited by restricted manufacturing rules. If there are unsolvable conflicts left, the fixing process has to go back to the routing stage to perform rip-up and re-route. Moreover, abutted placed cells may cause conflicts by patterns near the boundary. Standard cells are designed with various performance considerations, and have pre-defined pin locations. Although layout modification technique [41, 89] can be applied to separate those closed patterns, the moving space inside a cell is limited and the performance impact is questionable. Therefore, a larger loop back to the placement stage would be necessary to fix conflicts caused between cell boundaries. The turnaround time to fix coloring conflicts can be huge if the decomposability issue is not addressed in the design flow.

There are two directions to integrate SADP awareness into the placement stage: pre-defined coloring and on-the-fly coloring. Since the layouts of standard cells are known, we can perform layout decomposition up front and embed the coloring information in the cell library that can be used by the placer (pre-defined coloring). The alternative is to perform coloring when cells are placed (on-the-fly coloring). This approach may provide the most comprehensive coloring choices, but the repeated coloring would be time-consuming. In the following, we will focus on the approach based on pre-defined coloring.

2.2.1.2 SADP Challenges for Standard Cells

It has been mentioned that conflicts may occur in the boundary of two abutted cells. Standard cells contains power/ground net as well as signal nets, where power/ground net is connected throughout the entire row and thus is viewed as a single pattern. While conflicts between signal nets can be eliminated by simply moving the two cells apart, conflicts between power/ground net is hard to solve. Fig. 2.5 shows the coloring results of two cells, where the colors of the power and ground nets are different in (a) and are the same in (b). Since the coloring results in Fig. 2.5 are the only options for the two cells because of their internal pattern structures, a conflict exists natively when they are placed on the same row. For other DPL processes such as LELE-type DPL, this conflict may be solved by splitting the conflicting pattern into



(a) (b)

Figure 2.5: Layout decomposition conflicts on power/ground net. (a) AOI21. (b) NAND2.

two. However, this technique does not work for SADP because of the process limitation.

The overlay problem should be taken care when placing together two cells with pre-define colors. Fig. 2.6 shows two cases when two cells are placed, where red patterns are formed by the mandrel mask and blue pattern are formed by spacer and the trim mask. Assume there is no conflict in the boundary of cell B and C in (a). However, the blue patterns near the boundary would suffer from the overlay error because there is no adjacent mandrels to provide spacer alignment. Although we could reserve empty space between cell B and C to allow dummy mandrel insertion, this would increase the design area and wire length, and sacrifice the benefit of using DPL technology. Fig. 2.6 (b) shows the result with less overlay error by flipping cell C, where the mandrel in cell C provides spacer alignment in the abutting boundary for cell B.

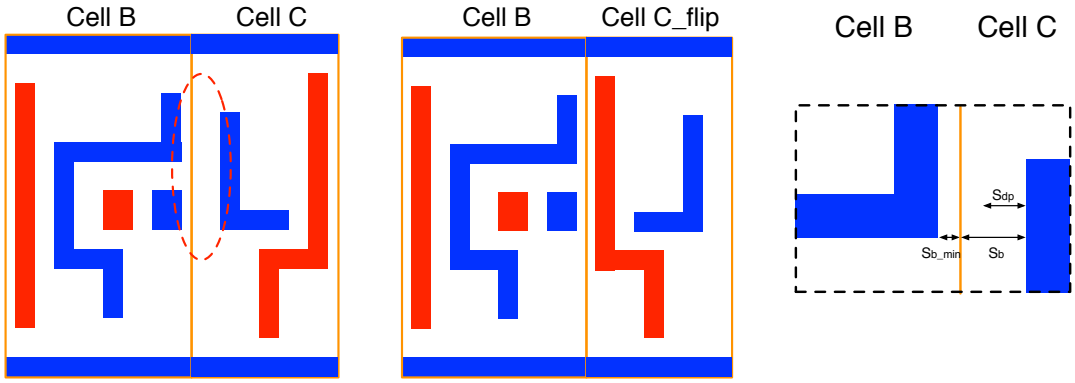


Figure 2.6: Placement example where (a) suffers more overlay error than (b). (c) Closer view of the center area of (a).

2.2.2 SADP-aware Legalization

The main challenge of SADP friendly placement is the extra effort to determine the decomposability during the placement stage. In addition, the design performance would be compromised because the decomposability becomes one of the optimization objectives for the placer. The problem can be even complicated when SADP overlay minimization is also considered. Therefore, an efficient approach is needed to make SADP friendly placement possible; and in the mean while, the impact to the design performance should be minimized. We present a SADP-aware legalization applied after the regular detailed placement. We will first give the problem formulation, and discuss what SADP-aware configuration should be provided in the cell library. Then we will explain how to integrate the configuration into the placement stage efficiently.

2.2.2.1 Problem Formulation

Given a placed layout, our task is to identify SADP conflicts between cells, and solve them by cell flipping or cell spreading. The objective is to solve all conflicts while minimizing the core area increase and wire length perturbation. Since we only care about the conflicts between cells, we assume a cell itself is decomposable, meaning there is no conflict internally. In addition, we assume "double-back" rows (adjacent rows share power/ground rail) are not used in the design for more flexible decomposition results.

2.2.2.2 Standard Cell Category

Before checking and solving conflicts in the placement, we need to build pre-coloring results and embed this information in the cell library. For each cell, we build a conflict graph to represent the topological relationship among pattens. Fig. 2.7 (a) shows a sample layout and its corresponding conflict graph. We traverse the conflict graph in DFS manner and apply two-coloring to assign a color for each pattern. Note that the coloring assignment may not be unique. To achieve the most placement flexibility, we enumerate all possible coloring candidates for each cell as shown in Fig. 2.7 (b)-(e).

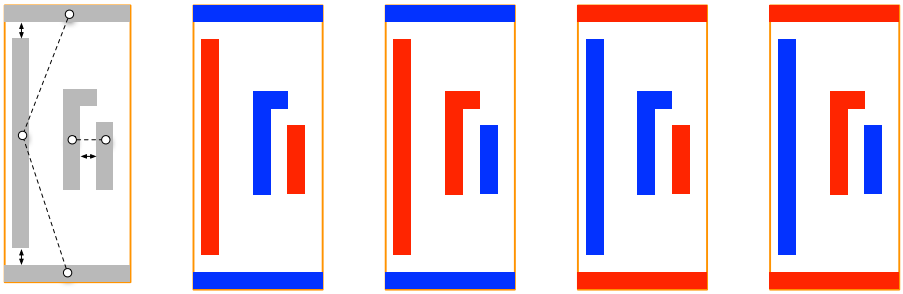


Figure 2.7: (a) Sample layout and its corresponding conflict graph in dashed lines. (b)-(e) Candidate coloring results.

Although we can always construct a conflict graph for two abutted cells and check if there is coloring conflict on the graph, this does not give much insight on how cell configurations impact decomposability. We further study the cell coloring results and categorize cells as PG-type and Abut-type which are defined as follows.

DEFINITION PG-type: The coloring relation between power and ground nets. PG-type can be (1) Same-PG if power and ground nets must be assigned the same color; (2) Diff-PG if power and ground nets must be assigned different colors; and (3) Free-PG if there is no coloring dependency between the power and ground nets.

DEFINITION Abut-type: The coloring and pattern geometry of a cell boundary (left and right). Abut-type can be (1) Safe-Abut if the spacing between the left/right-most pattern and the cell boundary is larger than the conflicting threshold S_{th}; (2) Free-Abut if there is no coloring dependency among the left- and right-most patterns, the power net, and the ground net; (3) Unknown-Abut if none of (1) or (2) is satisfied.

Examples of Diff-PG and Same-PG can be seen in Fig. 2.5 (a) and (b), respectively. Free-PG exists when there is no path connecting the power and ground net on the conflict graph, so they can be assigned either the same color or different colors. Cell A in Fig. 2.3 shows an example with Free-PG type, where a wide space between power/ground net and signal nets break their coloring dependency. It is obvious that a Same-PG cell cannot abut a Diff-PG cell; while a Free-PG cell is flexible to abut cells with any PG-type.

The left boundary of Cell C in Fig. 2.6 (a) is an example of a Safe-Abut boundary. Assume there is a minimum pattern-to-boundary spacing S_{b_min} between patterns inside a cell and the cell boundary as shown in Fig.

2.6 (c). We can find a threshold distance S_{th} such that as long as the pattern-to-boundary spacing S_b of a cell is larger than S_{th}, no conflict will be induced. This threshold is determined by the resolution limit, that is, $S_b > S_{th} = S_{dp} - S_{b\_min}$. The concept of Free-Abut is similar to Free-PG, which means the coloring of patterns on the boundary are independent and have nothing to do with the power/ground net. Patterns on a Free-Abut boundary can be colored freely depending on the color of its adjacent cell. For example, the right boundary of the cell in Fig. 2.7 is a Free-Abut boundary, while its left boundary is Unknown-Abut because the color of the left-most patten contradicts the color of the power/ground net.

Lemma 1: *Two cells are PG-compatible if and only if there is no conflict between their power/ground nets, including the combinations* {*Same-PG, Same-PG*}, {*Diff-PG, Diff-PG*}, *and* {*Free-PG, Any*}.

Lemma 2: *Two cells are Abut-compatible if and only if there is no conflict between patterns by their abutting boundary, including the combinations* {*Safe-Abut, Any*}, {*Free-Abut, Any*}, *and conflict-free* {*Unknown-Abut, Unknown-Abut*}.

Theorem 1: *Two cells are compatible if and only if they are PG-compatible and Abut-compatible.*

We define SADP compatibility of a pair of cells as the decomposability when they are placed abutted, which can be determined by Theorem 1. Determining PG-compatibility is trivial, however, determining Abut-compatibility

may requires extra effort if two Unknown-Abut boundaries are considered. In that case, we need to construct a conflict graph for patterns of two abutted cells and check if odd cycles (conflicts) are formed in the graph.

2.2.2.3 Decomposability Look-up Table

Although determining PG-compatibility is trivial, determining Abut-compatibility requires extra effort when Unknown-Abut boundaries involve. It is inefficient to perform coloring for Unknown-Abut boundaries whenever they are checked. Since cell library has fixed layouts and usually contains only hundreds of cells, we can create a decomposability look-up table (DPLUT) to provide quick query during legalization.

Given a standard cell library with N cells, we build a two-dimensional $N \times N$ DPLUT based on Theorem 1, in which each table entry keeps the decomposable solution candidates of two abutted cells. The first dimension represents the cell on the left, and the second dimension represents the cell on the right, as shown in Fig. 2.8. The solution candidates indicate decomposable cell orientations and the corresponding overlay error on the abutting boundary. If there is no legal combination to place two cells, their solution candidates would be $NULL$. Note that the same cell orientation may have different SADP layout decomposition results based on how patterns are colored, and we only keep the one with the minimum overlay error. With DPLUT, we can quickly query if two cells can be put together, and obtain the minimal overlay orientation as their placement solution. For example, the solution candidates

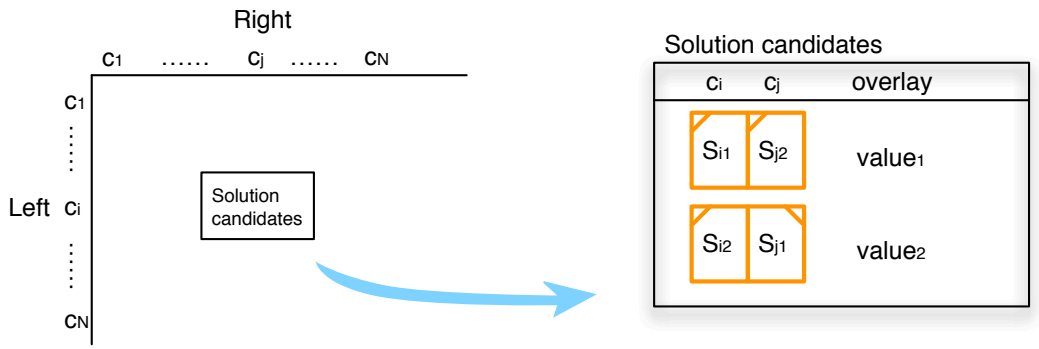


Figure 2.8: Decomposability look-up table.

in Fig. 2.8 indicate two orientations to place cell c_i and c_j, and thus we can decide wether to perform cell flipping according to the possible overlay error $value_1$ and $value_2$. Solutions S_{i1} and S_{i2} represent two different coloring results for cell c_i.

2.2.2.4 SADP Legalization

We present a post processing in detailed placement, SADP legalization, to "legalize" conflicting cells after the regular placement without significant design perturbation. After obtaining the DPLUT mentioned in Section 2.2.2.3, we use it to quickly determine the decomposability between two adjacent cells. Once a conflict is found, we applied one of the two techniques, cell flipping and cell spreading, to resolve it. When the design area is the main concern, cell flipping is preferred because it does not impose area overhead. Cell spreading may take advantage of existing white space on the original placement, but the white space may not be sufficient to solve all conflicts. In certain placement, we may need to enlarge the design area in order to resolve all conflicts.

Alg. 1 shows our greedy-based legalization. Rows of the placement are processed one by one, left to right (Line 1, 2). For each pair of conflicting cells, we first check if the conflict can be solved by flipping one cell as shown in Line 4. If a flipped solution is not available, then we try cell spreading as shown in Line 9. Decomposability check "IsConfclit" is done by checking if $\{c_{r_i}, c_{r_{i+1}}\}$ in the current orientation exists in DPLUT$[c_{r_i}][c_{r_{i+1}}]$. If not, we may flip one of the cells according to the solution candidates in DPLUT$[c_{r_i}][c_{r_{i+1}}]$ and the current color assignment of the two cells. However, if there is no any way to decompose the two adjacent cells, we can only solve their conflict by spreading the two cells.

Algorithm 1 SADP legalization

Input: R rows of cells

```
1:  for each r ∈ R do
2:    for i = 1 → |r|−1 do
3:      if IsConflict(c_{r_i}, c_{r_{i+1}}) then
4:        Flip(c_{r_i}, c_{r_{i+1}})
5:      end if
6:    end for
7:    for i = 1 → |r|−1 do
8:      if IsConflict(c_{r_i}, c_{r_{i+1}}) then
9:        Spread(c_{r_i}, c_{r_{i+1}})
10:     end if
11:   end for
12: end for
```

2.2.3 Experimental Results

Existing placement benchmarks such as ISPD 06' benchmark only provide placement information without standard cell library detail. Therefore, those benchmarks cannot be used for SADP legalization. Instead, we synthesize OpenSPARC T1 designs with Nangate 45nm standard cell library [5] to generate our benchmark. For simplicity, we assume the sizes of the minimum pattern width, spacing, and spacer width are the same, and modify the layout accordingly. Cells are decomposed as explained in Section 2.2.2 and used to configure the decomposability look-up table. Because Nangate standard cell library is not designed for SADP, several cells are not decomposable internally. For simplicity, we assume there is no internal conflicts and no power/ground incompatibility, so we can focus on solving conflicts between cell boundaries.

2.2.3.1 SADP Legalization

We perform placement with Cadence SOC Encounter [1] and use the result as the input of our approach. The default core utilization rate is set as 0.7. The benchmark information and our results are shown in Table 2.1.

We implement two versions of SADP-legalization, area-unbounded (UB) and area-bounded (B). In SADP-legalization_UB, expanding layout area is allowed for cell spreading if necessary. Table 2.1 shows the results, where all conflicts in the original placement are solved with slight area and wire length perturbation. On average, SADP-legalization_UB induces 3.25% additional area and 1.39% additional wire length.

Table 2.1: Experimental results with area-unbounded SADP legalization.

Design	Benchmark Statistics			SADP-Legalization_UB				
	Conflict	Area(um^2)	WL(um)	Conflict	Area(um^2)	WL(um)	+Area%	+WL%
alu	877	5284	29620	0	5451	30004	3.16%	1.30%
byp	2089	18011	133500	0	18997	135635	5.47%	1.60%
div	1439	11860	55390	0	12785	56535	7.80%	2.07%
ecc	587	5046	23090	0	5225	23376	3.55%	1.24%
ffu	612	6493	27970	0	6564	28216	1.09%	0.88%
mul	5463	42139	205500	0	42224	207978	0.20%	1.21%
efc	454	4471	12150	0	4536	12326	1.45%	1.45%
Average							3.25%	1.39%

In SADP-legalization_B shown in Table 2.2, layout area is fixed, and conflicts can only be solved within the given area specification. Our results show that around 40% of the conflicts can be solved without any area penalty, and the wire length perturbation is only 0.08% on average. SADP-legalization for all designs can be accomplished within seconds, showing the efficiency of utilizing DPLUT.

Table 2.2: Experimental results with area-bounded SADP legalization.

Design	SADP-Legalization_B			
	Conflict	WL	-Conflict%	+WL%
alu	519	29648	40.82%	0.09%
byp	1433	133552	31.40%	0.04%
div	831	55437	42.25%	0.08%
ecc	364	23108	37.99%	0.08%
ffu	345	27991	43.63%	0.08%
mul	3133	205683	42.65%	0.09%
efc	267	12164	41.19%	0.12%
Average			39.99%	0.08%

2.2.3.2 Analysis of SADP-friendly Standard Cell Design and Placement

There are two aspects when talking about SADP-friendly standard cell design. One is internally SADP-friendly, meaning the cell itself is decomposable. Another is externally SADP-friendly, meaning the cell can easily abut on another cell without conflicts. It is essential to ensure standard cells are self-decomposable to achieve basic layout decomposability. It is also important to maintain external SADP friendliness, which can further improve placement

results. We analyze the layout after applying SADP-legalization, and observe some important factors that affect the result quality. Below we discuss some design strategies that can improve SADP-aware legalization. This information can be considered by cell designers and CAD engineer to better achieve SADP friendliness.

- The power/ground compatibility between cells is the biggest issue in SADP-aware placement. The cell library either needs to maintain consistent coloring of power/ground net for all cells, or it needs to provide both coloring options (the same or different coloring) for each cell to provide the most flexibility for decomposable placement.

- The patterns on the boundary of a cell is where conflicts may occur, and thus the pattern number on the boundary should be kept small and the pattern structure on the boundary should be as simple as possible. For example, if patterns on the boundary are assigned the same color, they can easily abut a cell with another color on the boundary. However, if the patterns are with mixed colors, finding the conflict-free match would be more difficult.

- Leaving white space would benefit SADP legalization. Most importantly, the white space should be evenly distributed among the rows or the entire layout to avoid the bottleneck of solving conflicts. The area increase or performance degradation is usually determined by the row or region that need the most white space for solving conflicts. If a particular row

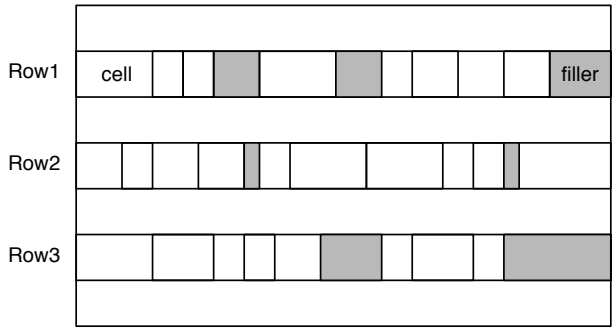


Figure 2.9: Row 2 may cause conflict solving bottleneck.

is much congested than others, the chance is that solving its conflicts needs to allocate more space by enlarging the core area. For example, Row 2 in Fig. 2.9 is more congested than the others; performing cell spreading for it requires either increasing the core area or moving cells in Row 2 to other rows with larger performance impact.

2.2.4 Summary

Double patterning enables nanometer lithography, however, achieving decomposable designs is still challenging. We study the coloring strategies of standard cells and analyze their impact to the placement with SADP process. We present the standard cell configuration and embed this information in the cell library. Based on the cell configuration, our SADP legalization can quickly determine the decomposability between cells and solve conflicts with cell flipping and cell spreading techniques. The results show that our approach can efficiently solve conflicts with small area overhead and layout perturbation.

2.3 SADP Aware Detailed Routing

In addition to standard cell layers that have been tackled for SADP in the previous section, lower routing layers are usually congested as well. To guarantee layout decomposability, it is necessary to consider SADP in earlier design stages, especially in detailed routing.

Because the most critical patterning control in SADP is not governed by lithography, but by the deposition of the sidewall spacer, it has less overlay error and excellent variability control compared to LELE [57]. However there is no stitch allowed in SADP to resolve conflicts, making its layout decomposition difficult. Moreover, SADP layout decomposition is not intuitive in the sense that the decomposition result does not have a direct relation to the original layout. SADP requires assist mandrels [9] during its patterning process and these unwanted mandrels need to be trimmed out by the trim mask. Therefore, the core mask and the trim mask cannot be obtained simply from the target layout. For 2D patterns, this mask assignment process would be more complicated.

Double patterning friendly routing has been proposed in [14, 88], but their methods cannot be applied to solve SADP induced issues. A SADP-friendly detailed routing flow [61] is presented by performing detailed routing and layout decomposition concurrently. In [48], a new grid structure with routing rules is presented and can be applied for SADP- and SAQP(self aligned quadruple patterning)-aware routing. The grid structure partially pre-assigns different colors to adjacent rows/columns, and the routing can be obtained by

connecting two pins on grids with the same color. To capture the decomposition violations and spacer-is-dielectric intrinsic residue issues, a graph model is proposed in [25] and a negotiated congestion based scheme is applied to solve the overall SADP routing problem. In order to further maximize the layout decomposability, we propose a systematic SADP-aware detailed router by integrating layer assignment and multi-layer routing structure to solve potential conflicts.

In this section we propose a robust multi-layer SADP-aware detailed routing algorithm which includes the following features:

- We propose a novel SADP-aware detailed routing approach that can handle 2D patterns on multi-layer designs in the presence of obstacles.
- We solve routing and layout decomposition simultaneously based on the correct-by-construction approach.
- We incorporate layer assignment to resolve potential pattern conflicts, which increases the flexibility of layout decomposition for SADP.
- We present a set of SADP-aware routing guidelines, which helps improve the pattern quality of SADP.

The rest of Section 2.3 is organized as follows. Our prescribed layout planning techniques are explained in Section 2.3.1. The details of the proposed SADP-aware routing framework are presented in Section 2.3.2. The

experimental results are discussed in Section 2.3.3, followed by the summary in Section 2.3.4. The preliminary results of this work were reported at [30].

2.3.1 Prescribed Layout Planning for SADP Compliance

Our objective is to achieve better SADP compliance by performing routing and SADP layout decomposition simultaneously. As a result, the routing solutions are able to take advantage of SADP's good overlay control. In this section, we present SADP-friendly routing guidelines to improve pattern quality and reduce decomposition conflicts.

2.3.1.1 SADP-aware Routing Guidelines

Mandrel patterns and trim patterns are fabricated by different manufacturing processes. The interaction between these two types of patterns may affect the printing images. Therefore, simply determining whether a layout is decomposable is not adequate for SADP-friendly routing. We analyze the impact of different pattern assignments on the pattern quality. The following three layout planning guidelines provide a systematic procedure to construct a SADP-friendly routing. Incorporating these guideline into our routing framework enables us to take advantage of SADP technology.

1. If both mandrel pattern and trim pattern are conflict-free when being assigned to a route, the mandrel pattern is preferred.

2. If the candidate routes have the same routing cost and can only be

assigned as trim patterns, the route with more spacer protection is preferred.

3. The distance between a trim pattern and a mandrel pattern is suggested to be larger than the forbidden spacing S_{forb}; although a valid routing solution only requires the minimum spacing $S_{dp} < S_{forb}$ to be satisfied.

These guidelines are explained below. The simulation result in [54] observes the printability degradation for the second mask lithography due to the presence of topography generated from the first mask on the wafer. One degradation can be seen from the CD variation, where patterns from the second lithography tend to have wider width when there are underlying patterns from the first litho/etch step. As a result, SADP prefers mandrel patterns from the first lithography for better printability control, and is different from LELE which prefers two balanced subsets of patterns [78].

Another advantage of SADP is the use of spacer. As illustrated in Fig. 2.2, when the boundary of a target pattern is aligned to a spacer, the overlay error can be prevented. Given this auto-alignment property of the spacer, a trim pattern protected by multiple spacers is preferred.

The minimal spacing S_{dp} in DPL constraints the minimum allowable distance between any two identical type patterns. A conflict occurs if two patterns within S_{dp} are assigned to the same mask. In addition, a forbidden spacing needs to be considered. The simulation results in [61] show that the printed image of a trim pattern would be affected by a close mandrel pat-

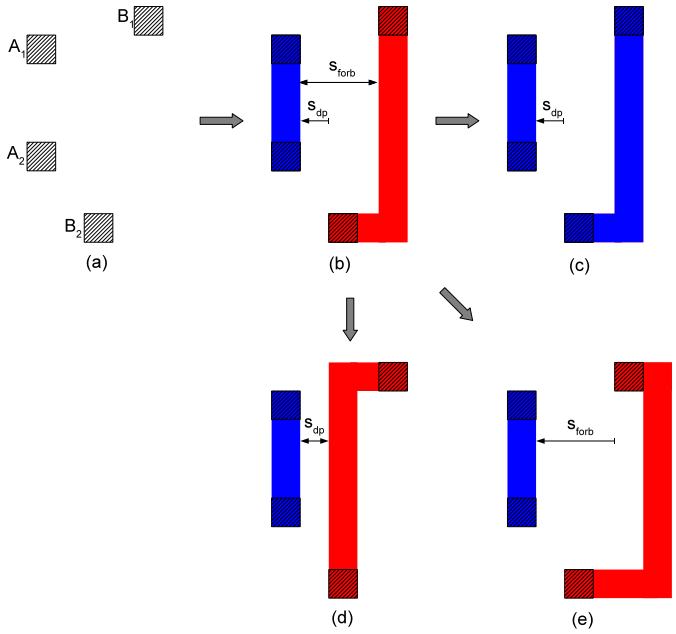


Figure 2.10: Prescribed layout planning. (a) Unrouted nets. (b) Legal patterns with bad quality. (c) - (e) Improved patterns by our prescribed layout planning.

tern even if S_{dp} is satisfied. In contrast, the quality of a trim pattern can be improved if its neighboring mandrel patterns are kept at a sufficient distance. Therefore, we define a forbidden spacing $S_{forb} > S_{dp}$ such that any distance $d_{mt} < S_{forb}$ is discouraged, where d_{mt} denotes the distance between a neighboring trim and mandrel pattern.

These layout planning techniques work as prescriptions for our routing engine to generate SADP-compliant layout patterns and to prevent patterns with bad quality. The example in Fig. 2.10 shows how routing patterns can be improved by our approach. The pin locations are given in (a) for two unrouted nets, and (b) is one routing and layout decomposition solution

without considering SADP. Mandrel pattern is shown in blue and trim pattern is shown in red in our following explanation. Although (b) is a legal solution by satisfying S_{dp} constraint, the mandrel pattern and the trim pattern may affect each other because their distance are within S_{forb}. Three alternative solutions with better pattern quality are shown in (c) - (e); where (c) adopts more mandrel patterns; (d) acquires more spacer protection; and (e) enlarges the distance between neighboring mandrel and trim patterns.

2.3.1.2 Simultaneous Layer Assignment for Conflict Prevention

The biggest challenge of SADP is the prohibition against using stitches. For a route $path_1$ on a single layer, either all grids in $path_1$ are assigned as mandrel patterns or all are assigned as trim patterns. This limitation dramatically decreases the possibility of generating a decomposable layout for SADP. In order to increase the flexibility of SADP layout decomposition, we perform simultaneous layer assignment during routing. In contrast to single-layer layout decomposition, multi-layer layout decomposition allows patterns to be assigned independently if they are on different layers. For example, a route $path_2$ is composed of seg_1- via_{12}-seg_2, where seg_1 is on metal 1, seg_2 is on metal 2, and via_{12} is used to connect seg_1 and seg_2. Since seg_1 and seg_2 are on different metal layers, they can be decomposed independently without introducing any conflict. Via can be viewed as a splitting point similar to the function of stitch in LELE layout decomposition.

Performing layer assignment during layout decomposition on multi-

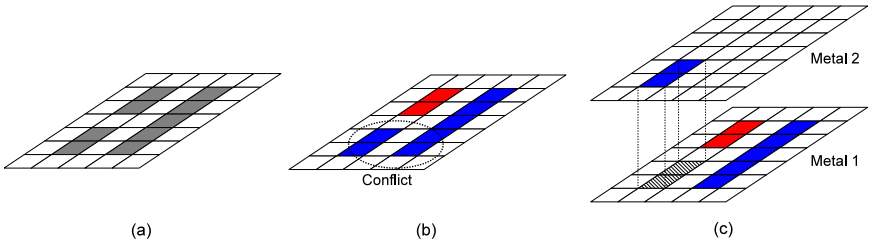


Figure 2.11: Prevent conflicts by simultaneous layer assignment. (a) Target layout. (b) Conflict occurs in single-layer layout decomposition. (c) Conflict removed by proper layer assignment.

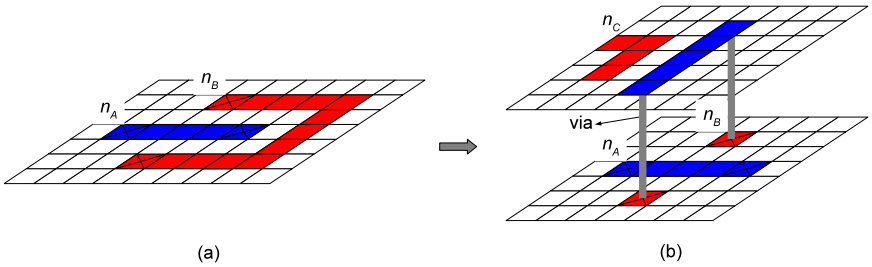


Figure 2.12: Increase flexibility of layout decomposition by simultaneous layer assignment. (a) Single layer. (b) Multiple layers.

layer designs has two advantages for SADP compliancy. First, a conflict can be easily resolved by assigning conflicting patterns into different metal layers. Fig. 2.11 shows a conflict that is solvable by our simultaneous layer assignment. Fig. 2.11(a) is the target layout that needs to be printed by DPL. A conflict occurs after single-layer layout decomposition in (b). By properly assign the patterns to different layers as shown in (c), the conflict can be prevented. The second advantage of considering layer assignment is that it increases the flexibility of layout decomposition. Fig. 2.12(a) shows an example after routing and layout decomposition on a single layer. Net n_B needs to detour to prevent intersecting net n_A. By assigning a section of patterns on n_B

to an upper layer as shown in (b), wirelength is reduced. Besides, the patterns on different layers are not restricted to a single color. In Fig. 2.12(b), patterns of n_B on metal 1 is assigned as trim patterns (shown in red); while the pattern on metal 2 is assigned as mandrel pattern (shown in blue) to provide spacer protection for the routed net n_C and to prevent conflicts.

The simultaneous layer assignment technique increases the solution space of both routing and layout decomposition, and thus helps prevent conflicts. This layer assignment is integrated into our three-dimensional path finding process, which will be explained in the next section.

2.3.2 Multi-layer SADP-aware Detailed Routing

This section gives the detail of our proposed routing framework. We first introduce the overall flow, and then present the techniques incorporated in the flow.

2.3.2.1 Overall Flow

We adopt a correct-by-construction approach to build our routing flow. When a net is routed, its layout decomposition is done simultaneously. During path finding, a rule checking procedure ensure not only a route is legal but also its patterns are decomposable. Consequently, once the routing is done, its layout decomposition result is also obtained.

Alg. 2 explains the overall flow of our approach. First, we perform initial layout decomposition for the blockages composed of pre-routed nets.

Algorithm 2 SADP-aware detailed routing

Input: A set of blockages B, and a set of nets N

1: Layout decomposition for B
2: $Q \leftarrow$ An arbitrary net $n_{begin} \in N$
3: **while** $!Q.empty()$ **do**
4: $n \leftarrow Q.pop()$
5: **for** each 2-pin net $k \in n$ **do**
6: Three-dimensional A* search for k
7: **end for**
8: **for** each $n_{neighbor} \in N$ where bbox of $n_{neighbor}$ overlaps bbox of n **do**
9: $Q \leftarrow Q + n_{neighbor}$
10: **end for**
11: **end while**

Since pre-routed nets in this stage are usually sparse, most would be assigned as mandrel patterns according to Guideline 1. Next, we process the input nets sequentially according to the routing order determined in line 8-9 (Section 2.3.2.3). Each multiple-pin net is decomposed into 2-pin nets and then routed using our three-dimensional A* search in line 5-7 (Section 2.3.2.4). The routing cost in A* search is a combination of wirelength and SADP cost, which will be illustrated in Section 2.3.2.2. After the A* search, the pattern assignment with lowest cost will be chosen.

2.3.2.2 SADP-aware Weighted Cost

When performing A* search, the cost of routing on a grid is calculated. Suppose an edge connecting grid g_i to g_j is considered, the cost of routing g_j as a mandrel and as a trim pattern is defined as follows:

$$\begin{cases} cost_j(m) &= cost_i(m) + \alpha \cdot WL_{ij} + \beta \cdot SADPC_j(m) \\ cost_j(t) &= cost_i(t) + \alpha \cdot WL_{ij} + \beta \cdot SADPC_j(t) \end{cases} \tag{2.1}$$

if g_i and g_j are on the same layer.

$$\begin{cases} cost_j(m) &= \min\{cost_i(m), cost_i(t)\} + \\ & \quad \alpha \cdot WL_{ij} + \gamma \cdot VIA + \beta \cdot SADPC_j(m) \\ cost_j(t) &= \min\{cost_i(m), cost_i(t)\} + \\ & \quad \alpha \cdot WL_{ij} + \gamma \cdot VIA + \beta \cdot SADPC_j(t) \end{cases} \tag{2.2}$$

if g_i and g_j are on different layers.

The pre-calculated cost $cost_i(m)$ and $cost_i(t)$ represent the cost when g_i is assigned as a mandrel pattern and a trim pattern, respectively; WL_{ij} is the wirelength between neighboring grids g_i and g_j; VIA is the via cost and $SADPC$ can be either positive or negative to represent a bad or good impact on pattern quality, respectively. User-defined parameters α, β and γ adjust the weight between wirelength and SADP awareness. As mentioned previously, stitch is not allowed in SADP. Therefore, g_j must be assigned as the same pattern of g_i if they are on the same layer, just as defined in Eq. (2.1). When multi-layer designs are involved, more optimization options are available. Therefore Eq. (2.2) provides more solution space when searching on multiple layers.

$SADPC$ is the double patterning cost when a grid is assigned as a mandrel/trim pattern and is determined by the guidelines provided in Section 2.3.1.1,which is defined as follows:

$$SADPC = \begin{cases} C_{mandrel} \\ C_{trim} \end{cases} - m \cdot C_{spr} + n \cdot C_{forb} \tag{2.3}$$

$C_{mandrel}$ and C_{trim} are the unit cost of assigning a grid as a mandrel or a trim pattern, respectively. The weight of $C_{mandrel}$ is set to be less than

the weight of C_{trim} according to Guideline 1 such that more mandrel patterns will be used. C_{spr} represents the benefit of a self-aligned spacer and thus it reduces the total $SADPC$ according to Guideline 2. The number of newly generated spacer-protected grids m can be optimized by routing more mandrel patterns next to existing trim patterns, or routing more trim patterns next to existing mandrel patterns. C_{forb} represents the penalty for patterns violating W_{forb} according to Guideline 3. Similar to m, n is the total number of newly generated forbidden grids by the current routing path. Note that violating W_{forb} is not encouraged, but it is valid for double patterning.

In general, the weight of these SADP costs differs depending on the technology. However, we may adjust the weight according to the routing density. For example, a larger C_{spr} encourages the binding of mandrel and trim patterns, and thus helps generate a tighter layout. In contrast, larger C_{forb} encourages a detour to prevent violating forbidden spacing, and thus consumes more routing resources. In our experiments, we set $C_{mandrel}$=C_{spr}=C_{forb} and C_{trim}=2$C_{mandrel}$.

2.3.2.3 Neighborhood-based Net Ordering

How a routing algorithm explores its solution defines how important net ordering is. For an ILP-based algorithm, solutions are calculated currently, thus net ordering is unnecessary. However, ILP-based algorithms usually have high runtime overhead. On the other hand, a sequential routing algorithm that processes nets one by one relies on a good net ordering method. The

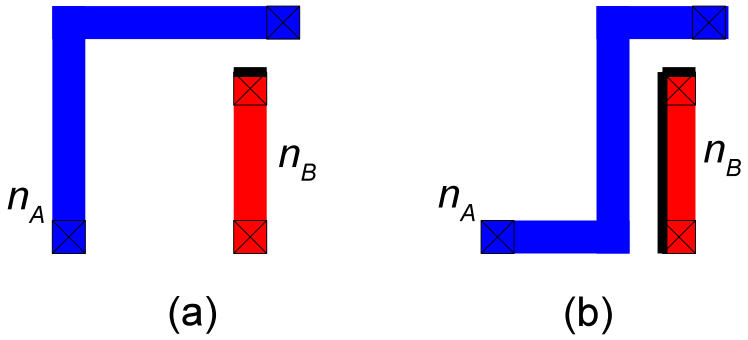


Figure 2.13: Net ordering impact on pattern quality. Bolder lines show grid boundaries that are protected by spacers. (a) Net n_a is routed first. (b) Net n_b is routed first.

better the net ordering is, the less rip-up and reroute are required and the less the runtime is needed. According to the cost function defined in Section 2.3.2.2, a preferred routing path should keep a low wirelength and has more spacer-protected grids. Fig. 2.13 shows the comparison of a bad and a good net ordering. In (a), net n_A is routed first and then n_B is routed. The bold line in the grid boundary shows where the grid boundary is protected. The net order of Fig. 2.13(b) is contrary to (a). We can see that with the same wirelength, the solution in (b) obtains much more spacer protection.

To achieve SADP-friendly net ordering, we propose an ordering method based on the geographic relation among nets. First, an arbitrary net n_i is selected to be routed. After n_i is routed, we obtain the next net to be routed n_j by finding every $bbox_{n_j}$ overlapping $bbox_{n_i}$. Here $bbox_n$ is determined by enlarging the net bounding box by a specific width w_{enl}. This ordering method encourages nets within a certain distance to be routed in a sequence, so that

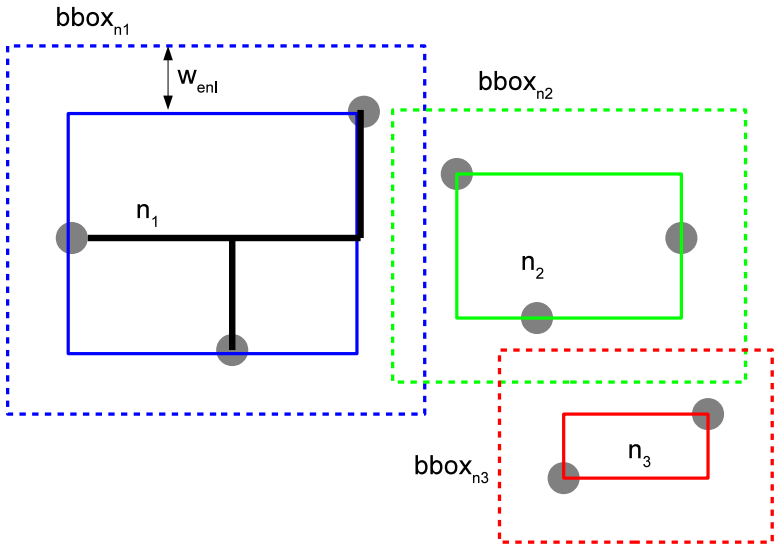


Figure 2.14: Neighborhood-based net ordering. n_2 allows more spacer protection to be provided for n_1.

the probability to provide spacer protection for these neighboring nets can be increased. In our implementation, we set w_{enl} slightly larger than S_{forb} so that the enlarged area is sufficient but not causes too much computational burden.

Fig. 2.14 shows an example of neighborhood-based net ordering. In the beginning, net n_1 is routed and the next routing net will be determined. It can be seen that $bbox_{n_2}$ overlaps $bbox_{n_1}$ and thus n_2 will be routed next. Finally, n_3 will be routed because its $bbox_{n_3}$ overlaps $bbox_{n_2}$.

Because the searching for overlapping *bbox* needs to be done whenever a net is routed, it is important to reduce the overhead of this search. We adopt R-tree [39] for fast indexing *bbox* information.

2.3.2.4 Efficient Three-dimensional Path Finding by Dynamic Programming

During path finding, when a routing grid g is considered, the validity of assigning g as a mandrel pattern (blue) and a trim pattern (red) is checked simultaneously. The combined routing and layout decomposition result is denoted as $R(path, LD(path))$, where $path$ is the routing path composed of grids, and $LD(path)$ is the coloring result for $path$. If a solution candidate $R(path_1, LD(path_1))$ generates any conflict, a high routing cost defined in Section 2.3.2.2 would be applied to prevent this candidate being selected.

The solution space for $R(path_i, LD(path_i))$ $\forall i$ in single-layer SADP is limited because all grids $g_j \in path_i$ must be assigned as the same color. However, the solution space on multi-layer designs would be much larger. As discussed in Section 2.3.1.2, simultaneous layer assignment with routing enables more flexible layout decomposition. Therefore, we adopt a three-dimensional path finding so that layer assignment can be integrated into the routing process. Fig. 2.15 shows a routing path connecting pins p_1 and p_2. Because the path is composed of three independent segments, seg_1, seg_2, seg_3, which are connected by vias, each segment is flexible to be assigned as either a mandrel or a trim pattern. It can be seen that in total 8 candidate solutions are available for the case in Fig. 2.15.

The time and space complexity would be an issue if we simply explore all possible solutions during three-dimensional path finding. We find that, in fact, it is not necessary to maintain all combination of $R(path_i, LD(path_i))$

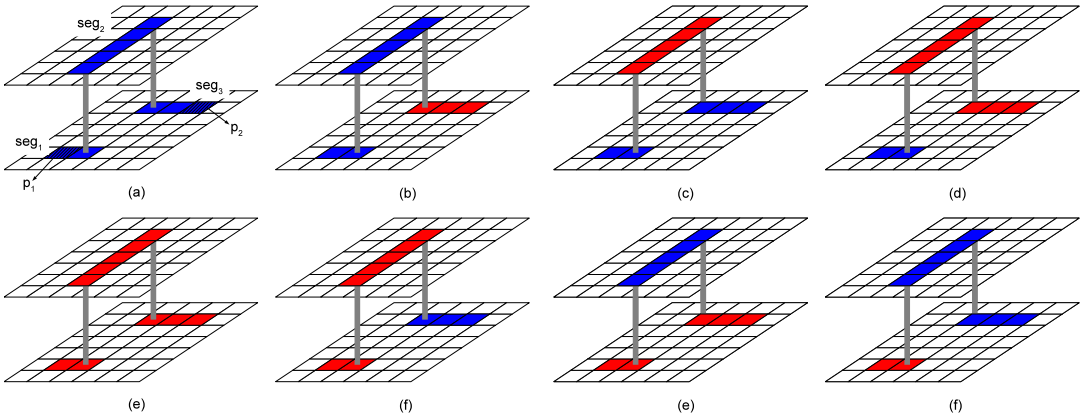


Figure 2.15: Solution candidates for multi-layer SADP.

during simultaneous routing and coloring. Given this observation, we develop an efficient three-dimensional path finding based on dynamic programming.

Assume a grid g_i is considered to be routed by a 2-pin net $n(g_s, g_t)$ where g_s and g_t are the source and sink pins, respectively. We first evaluate the costs of assigning g_i as a mandrel pattern and as a trim pattern. According to the definition in Eq. (2.1) and (2.2), we then obtain the accumulated cost along the path from g_s to g_i. Although there are many solution candidates for the routing path through g_i, we only need to maintain two solutions, $cost_i(m)$ and $cost_i(t)$, where $cost_i(m)$ and $cost_i(t)$ are the accumulated routing costs when g_i is assigned as a mandrel pattern and a trim pattern, respectively. By keeping the minimum $cost_i(m)$ and $cost_i(t)$ in $path_{s,i}$ for each traversed grid g_i, we are guaranteed to obtain the minimum cost solution for $path_{s,t}$. The solution for the routing path of $n(g_s, g_t)$ can be expressed as the following recursive form of dynamic programming:

$$R(path_{s,t}, LD(path_{s,t})) = \\ R(path_{s,i}, LD(path_{s,i})) + R(path_{i,t}, LD(path_{i,t})) \tag{2.4}$$

, for any g_i in the routing grid

According to Eq. (2.4), we only need to maintained two minimum cost solutions $cost_i(m)$ and $cost_i(t)$ for any grid g_i traversed during A* search. This makes our three-dimensional path finding more efficient on both time and space.

2.3.3 Experimental Results

We implemented the proposed algorithm in C++ and tested it on the machine with 2.66GHz CPU and 4G memory. The parameters in Eq. (2.1) and Eq. (2.2) are set as follows: $\alpha = \beta = 1$ and $\gamma = 0.3$. Two experiments test the performance and robustness of our approach. The first experiments contains only single-layer and obstacle-free designs, while the second experiment includes multi-layer designs in the presence of obstacles. For single-layer design, the method in [61] is implemented and compared with our approach. For multi-layer designs, our results are compared with a wirelength-driven routing method.

First we compare our result with [61] which simply works for single-layer designs. Because [61] also adopts A* search technique, we are able to incorporate its cost function into our routing flow. However, due to the unavailability of the benchmark in [61], we randomly generate test cases to perform the comparison. Four cases are generated with different number of nets

as shown in Table 2.3. Note that the layout size of these cases is the same; in which Case1 has the lowest routing density while Case4 has the highest routing density. We compare the result in terms of wirelength (WL) and double patterning performance including (1) the number of spacer-protected trim patterns (#SP-trim), (2) the number of non-spacer-protected trim patterns (#NSP-trim), (3) the number of forbidden grids (#FORB grid), and (4) the number of conflicts (#conflict). The result shows our approach consistently generates better pattern quality with only a 3% wirelength increase. On average, our result generates 51% more spacer-protected trim patterns than [61], in which spacer protection implies better pattern quality. In addition, we reduce the number of non-spacer-protected trim patterns and forbidden grids by 39% and 55%, respectively.

Table 2.3: Result comparison with [61] on single-layer designs.

Testcase	#Nets	Router	WL	Double Patterning			
				#SP-trim	# NSP-trim	# FORB grid	#conflict
Case1	300	[61]	3770	28	63	3	0
		Ours	3820	40	33	0	0
Case2	600	[61]	7258	209	346	26	0
		Ours	7330	250	216	12	0
Case3	800	[61]	9704	427	727	48	0
		Ours	10130	725	464	25	0
Case4	1000	[61]	12171	750	1107	122	0
		Ours	12929	1291	702	101	0
Avg Ratio			1.03	1.51	0.61	0.45	1

We then test the performance of our approach on multi-layer designs in the presence of blockages. Since there is no previous routing work taking double patterning into consideration on multi-layer designs, we implement a multi-layer wirelength-driven routing method followed by SADP layout decomposition as our comparison baseline. A set of two-layer industrial designs are scaled down to 22nm technology for the experiment. Table 2.4 gives the statistics of these designs. Each design contains two metal layers, M1 and M2, and blockages appear on both layers. Table 2.5 shows the comparison between our approach and the wirelength-driven routing in terms of wirelength, the number of vias (#Via), double patterning performance and runtime. Our approach achieves a great improvement in the results of double patterning. On average, the number of spacer-protected trim is increased by 2.87X; and the number of non-spacer-protected trim patterns and forbidden grids are reduced by 31% and 49%, respectively. The runtime of WL-driven is less than our approach because it does not perform any decomposability checking. It is worth mentioning that the benchmarks are quite dense and some areas contain congested pins which are difficult for double patterning technology. Table 2.5 also shows that unresolvable conflicts exist in both of our result and wirelength-driven result, which may be fixed by post-routing techniques. Our approach outperforms wirelength-driven routing with fewer conflicts. The number of vias is increased by 32% because we utilize layer assignment to prevent conflicts and to improve the pattern quality.

Table 2.4: Benchmark statistics for multi-layer designs.

Circuit	Size(um^2)	#Nets	#Blockages		
			M1	M2	Tot
CK1	20x20	29	279	26	305
CK2	48x48	306	3528	210	3699
CK3	100x100	872	13207	766	13813
CK4	160x160	1937	38792	2029	40370

Table 2.5: Result comparison of routing and layout decomposition on multi-layer designs.

Circuit	Router	WL	#Via	Double Patterning				Runtime(s)
				#SP-trim	# NSP-trim	# FORB grid	#conflict	
CK1	WL-driven	22911	48	320	262	13	22	2.3
	Ours	23045	60	480	179	3	15	6.6
CK2	WL-driven	126215	616	2397	5248	794	251	37.8
	Ours	133893	906	9397	3539	518	136	208
CK3	WL-driven	530555	1788	6222	10588	1772	757	190.8
	Ours	536215	2292	18162	7491	923	290	1021.6
CK4	WL-driven	1269046	4484	13740	25005	4375	1682	556.2
	Ours	1297775	5708	43238	17587	2787	670	2802.5
Avg Ratio		1.02	1.32	2.87	0.69	0.51	0.50	4.69

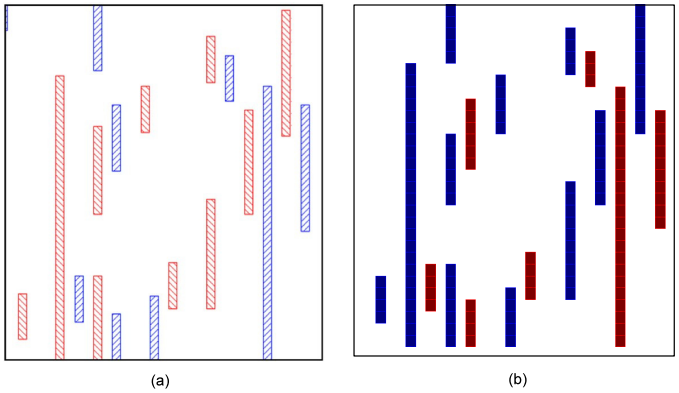


Figure 2.16: Sample Layout decomposition result by (a) [9] and (b) our approach.

Fig. 2.16 shows a 1D layout generated by SADP-friendly layout decomposition [9] and our approach. Our result tends to generate more mandrel patterns and reduces the number of non-spacer-protected trim patterns, which implies our result obtains better pattern quality according to the proposed routing guidelines.

Overall, our approach consistently achieves SADP-compliant results with negligible wirelength overhead. We provide more flexibility on layout decomposition by taking layer assignment into consideration. In addition, our prescribed layout planning techniques greatly improve the pattern quality and thus can benefit lithography manufacturing for SADP.

2.3.4 Summary

In this section, we propose a novel multi-layer SADP-aware detailed routing approach. A set of SADP-aware routing guidelines are presented,

which improves SADP compliancy. We adopt a multi-layer routing model and present simultaneous layer assignment to increase the flexibility of SADP layout decomposition. Our work simultaneously solves routing and layout decomposition problems using a correct-by-construction methodology. The experimental results show that the proposed approach achieves promising results on both single-layer and multi-layer designs.

2.4 SADP Compliant Post-Routing

Although embedding lithography-aware information in detailed routing as discussed in the previous section can achieve better DFM awareness, it involves significant addition of constraints and physical design time [19, 30]. Besides, the explosion of restrictive design rules may cause QoR (Quality of Results) degradation in terms of design metrics such as frequency, power and area.

In-design physical verification flow [7, 29] allows designers to configure an additional set of design rules that are not usually considered in typical routing flow, but are good for manufacturability. Violations caused by these rules are identified and then used to guide the routing engine to refine routes. Recently, Mann et al. presented a DFM optimization method [60] that adopts in-design flow. They perform hotspot fixing after obtaining the routing results, but only target at via replacement rules.

This section exploits a post routing approach that has the flexibility to resolve lithography violations without the overhead of repeated rule checking. In addition, it allows for successive refinement in the definition of lithographic violations as the process node matures, and implementation of fixes as localized ECO (Engineering Change Order) operations without needing to reroute the complete design. We employ in-design physical verification flow in a commercial router, which allows us to perform physical verification and pass the information to the router. Therefore, by configuring lithography friendly design rules for physical verification, the router can iteratively check lithography

validity and perform localized rip-up and reroute to fix violations.

The rest of Section 2.4 will be organized as follows. We discuss the routing challenges and our motivation in Section 2.4.1. We present our post routing flow in Section 2.4.2. The experimental results are discussed in Section 2.4.3, followed by the summary in Section 2.4.4. The preliminary results of this work were reported at [29].

2.4.1 Routing Challenges with Lithography Rules

A typical solution to avoid SADP-unfriendly layout consists of performing post-OPC lithography simulation and identifying the layout hotspots that lead to silicon failure. Unfortunately, the lithography simulation is time consuming, and therefore cannot be used to drive the routing engine. Another way is to correlate the model-based lithographic information to topological design rules that can be understood by the routing engine. However, the ability to route a given netlist within specified performance criteria (such as timing, current capacity, resistance, capacitance, etc.) in specified runtime constraints is inversely dependent on the number of design rules that need to be satisfied during routing. For sub-22nm node, the rule count is reported [59] up to 2000. Taking into account all design rules during the routing phase can be computationally expensive and may lead to performance degradation of the resultant layout. In practice, some important design specifications may be sacrificed in order to satisfy rules for manufacturability. Therefore, only a few selected design rules are considered during routing. However, the design rules ignored

during the routing phase can be significantly important to avoid lithographically difficult hotspots, thus leading to an adverse effect on the lithography quality of the resultant design.

Identifying the previously unconsidered design rules that caused lithographic hotspots can help successive application of selective design rules in routing without any degradation in circuit performance. For example, Fig. 2.17 shows the impact on lithography quality (LQ) and routability when the number of design rules increases. Lithography quality LQ(A) corresponds to the design choice A is obtained by considering only a few design rules during routing, and LQ(B) corresponds to the design choice B is obtained with larger number of design rules considered during routing. If many lithography design rules are considered, the lithography quality would obviously increase, but the routability would degrade. By using feedback from lithographic simulations

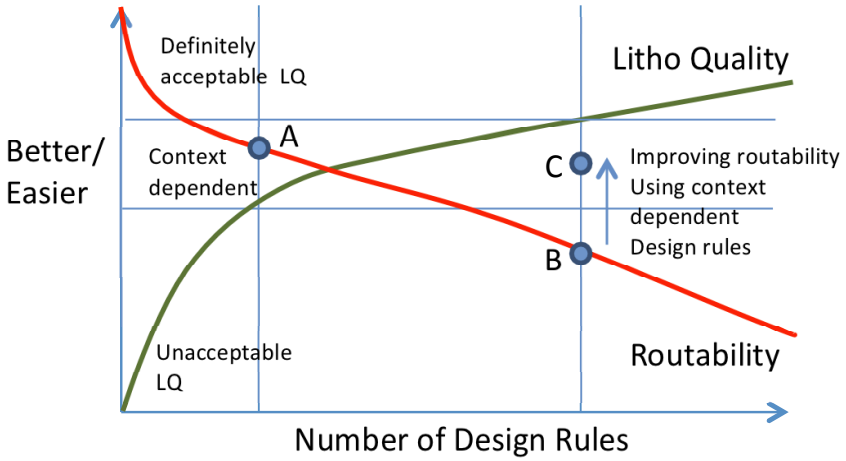


Figure 2.17: Lithography quality and routability based on different number of design rules.

and application of context-dependent design rules, the lithography quality of the design can be improved significantly from LQ(A) to LQ(C) without much loss in routability or circuit performance.

2.4.2 SADP-compliant Post-routing

We propose a new routing flow that allows SADP-compliant rip-up and reroute during post-routing stage. Fig. 2.18 shows our overall methodology, including two main steps: lithography rule extraction and lithographic hotspot fixing. First we perform lithographic simulation after typical routing flow. We then characterize the problematic patterns with properties that can be transformed into design rules. These rules are fed back to the routing engine where problematic patterns can be fixed in post-routing stage. The lithographic hotspot fixing is proceeded until an identified quality criteria is met or the iteration upper bound is reached.

2.4.2.1 Lithography-aware Design Rule Extraction

Advanced processes rely on model-based simulation to evaluate lithography quality accurately. However, it is difficult to adopt this approach in the optimization processes because it is computational expensive. Ignoring the lithography impact in the design flow clearly will create a gap between the obtained layout and the acceptable lithography-friendly layout. As an alternative, we analyze the simulation results under the process specification and transform the important factors into rules that can be applied by rule-based

approaches.

We perform lithographic simulation after typical routing flow considering only the mandatory design rules. Based on the simulation results, the analysis tool can identify faulty patterns according to process characteristics, including Edge- placement-error (EPE), variations of line-width and space, etc. We perform pattern matching that helps to classify hotspots caused by similar pattern topologies. The patterns that tend to cause larger number of hotspots are then selected and recommended as rules for improving lithography quality. Usually, they either have more restricted values for design specification, or involve particular features arrangement.

The main characteristic of these problematic patterns, including feature width, feature space, and the geometrical relation between features, are extracted and correlated to design rules. These rules are fed into the sign-off

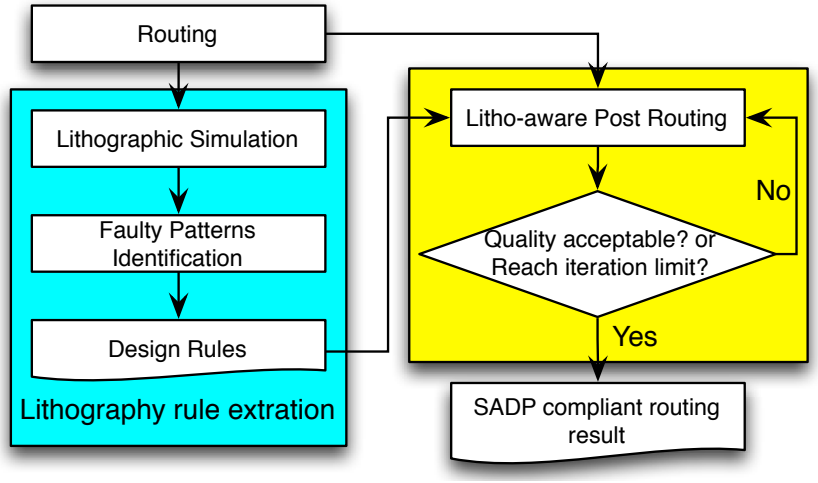


Figure 2.18: Overall methodology flow.

verification tool, which applies pattern matching to identify all faulty patterns in the design. The rules will also guide the routing engine to fix the identified patterns. Note that lithography rule extraction only requires one time effort for a particular manufacturing process/setting. Once the lithography-aware design rules are extracted, these rules can be generally adopted by designs with the same process.

The design rule extraction step is critical to identify the few layout structures leading to highest yield loss. The following cases must be considered:

- Inaccurate rules will result in useless layout structure matches that will be processed with the same priority of the real lithographic hotspots, and thus will waste routing resources and decrease the effectiveness of the fixing flow.

- A huge number of less critical layout structures can nevertheless impact yield. Therefore, the design rule extraction step should also include these hotspots, and the neighboring environment of the hotspot must be considered. The rule should identify layout topologies that could be fixed within the extracted environment, and the fix strategy must be adapted accordingly. As an example, the rule must capture the problematic pattern but also the context to enable correction with the minimum local layout changes.

- The fixing flow is implemented after detail routing and relies on the place and route information. Therefore, the macro and standard cell internals

cannot be modified. In addition, the rules must be designed to report the violation for the involved routing structure to avoid routing issues with the macro.

2.4.2.2 Lithographic Hotspot Fixing with In-design Physical Verification Flow

There have been several studies on SADP-aware routing as mentioned above, but there are some difficulties to adopt those approaches in real industrial design flow. First, previous SADP-aware routing studies define new routing strategies for the router to follow, which usually requires a fundamental change of the router behavior. This imposes an implementation overhead for routing tool. Furthermore, lithographic hotspots highly depend on the technology node, manufacturing process, and other parameters. It would be a huge burden to modify router implementation for different processes and foundry settings. Second, although considering SADP compliance during full routing provides large solution space for decomposable layout, it is computationally expensive to handle all rules. DRC and DFM rule count has been increased as the technology node shrinks. The router complexity to check these rules increases even faster because the rules are more complicated. Too much rules also restrict the solution space for other optimization, such as timing, power, etc. Third, foundries often provide recommended rules for manufacturability improvement except the mandatory DRC rules. These rules are nice-to-have, but are not enforced strictly. Therefore, these recommended rules should be given lower priority than DRC rules during physical design flow.

We adopt industrial in-design physical verification flow to integrate lithography awareness into routing stage. In- design physical verification flow performs concurrent physical design and physical verification, which helps to improve the turnaround time between physical design and physical verification. The concept is to integrate physical verification into routing engine and use verification results to guide the following rip-up and re-route. With this flow, we can formulate SADP-compliant rules as a part of sign-off physical verification, which performs pattern matching to identify violated layout structures.

Our methodology first performs regular routing without consider any manufacturing issues. The lithography-aware rules are then added into the signoff rule deck. We then apply physical verification to identify lithographic hotspots and to guide the localized rip-up and reroute. This process is iteratively performed until all hotspots are fixed or a given iteration count is achieved. The proposed flow has the following features:

1. Easy integration into the existing design flow. Since the SADP-aware rules are configured as signoff rules, there is no need to change the router implementation. These rules are described by formal semantics similar to DRC rules, which creates no ambiguity and can be easily modified depending on different process parameters.

2. Efficient SADP-aware routing. The SADP-aware rules are not considered in the main routing step, which avoids the runtime overhead for extra rule checking.

3. Prioritized rules checking. Our methodology only allows SADP-aware rules in post routing stage. Therefore, the router can first focus on the mandatory design rules and allows more optimization space in the main routing step. In addition, only problematic patterns will be rerouted, which avoids excessive layout changes to affect prior optimized results.

4. Inherent benefits with in-design flow. The in-design flow adopts accurate signoff physical verification with pattern matching, which is especially suitable for checking lithographic hotspots that are usually complicated. In addition, in-design flow can still consider timing closure that helps to keep design performance.

2.4.3 Experimental Results

The proposed methodology flow is tested on advanced process node designs. We first perform lithographic simulation on designs with the same technology. By observing patterns with bad printability, we extract their features and correlate them to design rules. Table 2.6 shows the results by feeding lithography-aware rules into our flow. Three rules are verified after the regular routing, and patterns that violate these rules are identified and re-routed locally. To prevent too much overhead for lithography-aware post routing, we limit the fixing iterations to 3. For each rule, we show the violation fix rate after all iterations. Note that these rules are considered simultaneously in each iteration.

Table 2.6: Post routing results after fixing lithography-difficult hotspots.

Design	Violation Fix Rate				Hotspot Red. Rate	△DRC	△WNS	△TNS	△CPU
	Rule 1	Rule 2	Rule 3	Total					
Design 1	81.20%	65.60%	71.70%	75.60%	58.70%	0	0	0	1.60%
Design 2	20.30%	55.20%	47.30%	24.00%	51.00%	-25	-1	0	30.00%

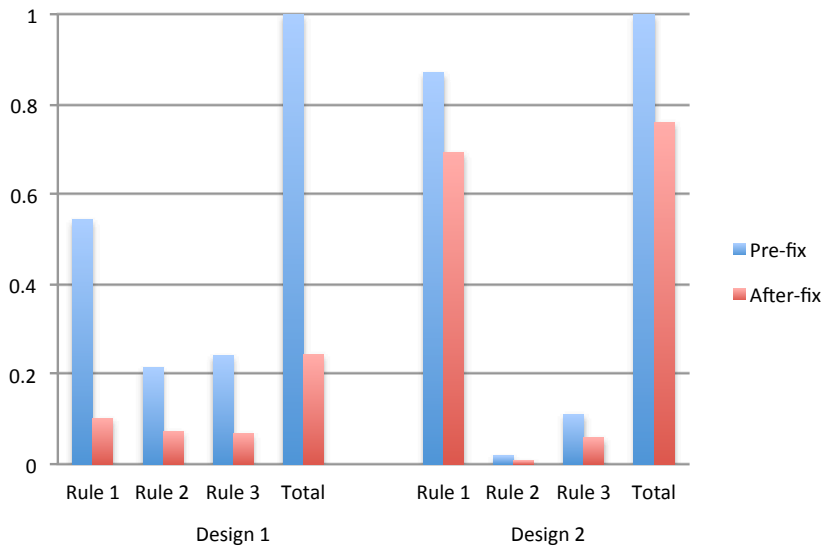


Figure 2.19: Normalized violation count for each rule.

The proposed flow works more effective for Design 1, where each rules has more than 65.6% violations fixed. Design 2 is larger and more complex, and thus the solution space for re-route is more limited. The breakdown of the violation percentages account for each rule is shown in Fig. 2.19. It can be seen that Rule 1 tends to identify more lithographic hotspots. The overall violation fix rate for Design 1 is 75.6%, while for Design 2 is 24%. Fig. 2.20 shows sample layouts before and after the hotspot fixing.

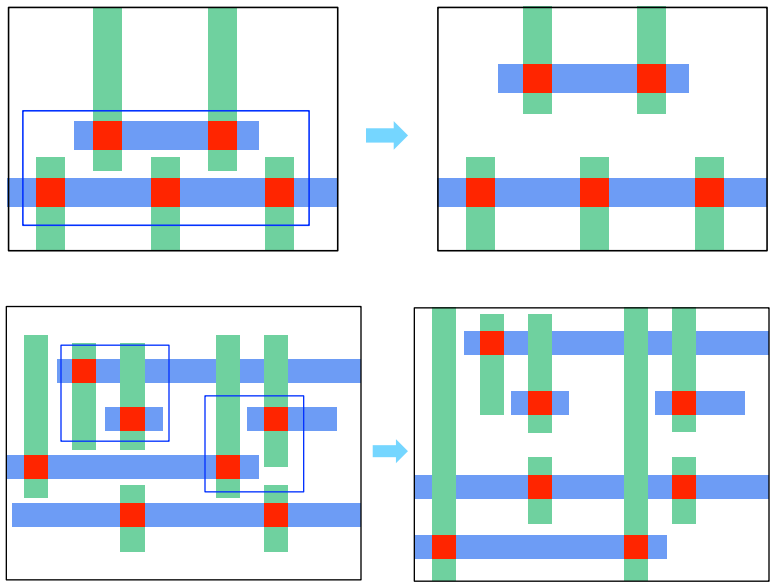


Figure 2.20: Sample layouts before (left) and after (right) fix. Blue boxes identify hotspots.

We further verify the impact of our lithography rules and re-routed solutions by performing the lithographic simulation. The hotspot counts are considerably reduced, where the hotspot reduction rate is 58.7 for Design 1, and 51% for Design 2. Fig. 2.21 shows the lithographic simulation results

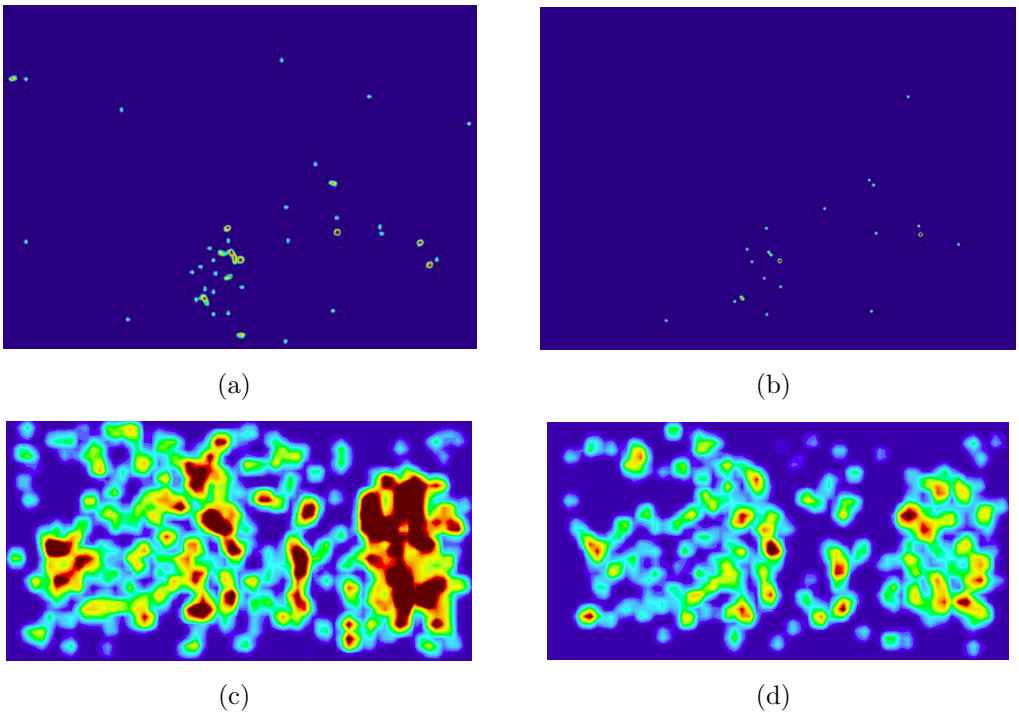


Figure 2.21: (a) Pre-fix and (b) after-fix hotspot density map of Design 1; (c) pre-fix and (d) after-fix hotspot density map of Design 2.

before and after the hotspot fixing for Design 1 and Design 2, where the hotspot density is much reduced after applying the proposed flow.

Note that these lithography-aware rules are not mandatory. Although our goal is to remove them as more as possible, it is okay not to remove them completely since they are not critical design rules. However, we should make sure the design retains its optimized state from prior routing stage, in terms of normal design rules, timing, etc. We collect the DRC report and timing report after applying our flow. Table 2.6 shows the difference of DRC ($\triangle$DRC), the worst negative slack ($\triangle$WNS), and the total negative slack ($\triangle$TNS). Our flow

does not degrade the timing performance. In fact, the timing of Design 2 is slightly improved after several rip-up and re-route. It is worth mentioning that no any violations are introduced in the mandatory design rules. The CPU time is reported as the additional post routing against the normal routing time. Although Design 2 is smaller than Design 1, its hotspot fixing time is much larger, reflecting more difficulty in finding valid routes.

The above results are obtained under fixed rip-up and re-route iterations for both Design 1 and Design 2. We perform another experiment to study the impact of iteration number. We observe that when the iteration number is doubled, the fix rate of certain rules is increased while that of the others is decreased. The overall fix rate by doubled iterations is even slightly worse than the results with less iteration. This shows that the fix rates in Table 1 has almost reached the upper bound for the given design space and rules. As a future study, we may prioritize these lithography rules according to their lithographic impact. For example, the most important rules are applied at the first iteration, and the other rules are gradually added in the following iterations.

2.4.4 Summary

We propose a SADP-friendly post routing methodology that adopts industrial in-design physical verification flow. Lithography-aware design rules are extracted from the lithographic simulation and are fed into the verification tool for hotspot detection. The identified hotspots can then guide the

localized rip-up and re-route. We compare the lithography quality between the typical routing flow and the proposed flow. The proposed methodology successfully reduces lithographic hotspots without introducing new violations for the existing design rules and without quantitatively impacting QoR of the design. Simulation results show that the hotspot reduction rate can be up to 58.7% compared to the design without considering lithography-aware rules. The lithography-aware design rules are treated equally and optimized simultaneously in this work. However, the lithographic impact of each rule and the difficulty to fix it may be different. As a future work, we would like to further study the importance of these rules and prioritize them during the iterative hotspot fixing to maximize the lithography quality.

2.5 SADP Layout Decomposition with Complimentary E-beam Lithography

When a layout has been generated through the physical design flow, it requires a layout decomposition step to determine the masks for double patterning lithography. Although with the aforementioned SADP-aware layout optimization approaches, finding a valid decomposition result may still be difficult as we keep pushing pitch scaling. Complementary lithography is an advanced technique that enables higher layout resolution.

Complementary lithography is proposed to allow 193nm optical lithography work hand-in-hand with high-resolution lithography to achieve advanced designs [11, 34]. In the first step, base features are created by cheaper optical lithography or self-aligned double patterning (SADP); in the second step, high-resolution lithography techniques are applied to cut unnecessary lines. Such line cutting can be accomplished by costly quadruple patterning, EUVL, or EBL. By carefully arranging how features are generated with the combined lithography techniques, we can achieve good pattern quality with a reasonable manufacturing cost. The advantages of adopting complementary lithography include: (1) high throughput by generating base patterns with mature optical lithography; and (2) improved mask yield by partial EUVL or EBL patterning, while no heavy manufacturing cost introduced.

Recently, the technique to combine optical and complementary EBL becomes promising. Lam et al. [50, 51] proposed using EBL to complement 193nm immersion lithography for 1D layout. As shown in Fig. 2.22, regular

lines are first fabricated by SADP; EBL is then used for line cutting to get the target layout. Although having their own advantage, standalone SADP and EBL are limited by low manufacturing flexibility and low throughput, respectively. Fortunately, by combining SADP with EBL together, we have the potential to achieve high productivity and pattern quality at the same time.

Several studies [50, 51, 62] have presented the effectiveness of applying SADP with line cutting technique on 1D layout designs. In order to optimize the overall throughput with hybrid SADP and EBL, an integer linear programming (ILP) -based approach [26] was proposed by properly distributing cutting patterns to the optical mask and e-beams. However, this work only targets at 1D gridded designs and allows wire end extension that is not always permitted for general designs. There have been some studies [9, 61, 74, 90] presented for pure SADP layout decomposition of 2D random patterns, where layout decomposition is the process to assign layout features into two different fabrication steps. These approaches impose strict SADP process rules to ensure the decomposed layout is SADP-manufacturable. However, the design

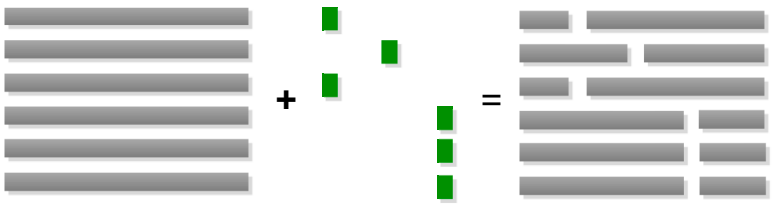


Figure 2.22: Complementary lithography for 1D layout.

flexibility is highly restricted and the layout may still not be decomposable given a complex layout.

In this section, we solve 2D layout decomposition problem that enables SADP with complementary EBL. To the best of our knowledge, there is no existing study considering SADP with complementary lithography on 2D designs. In addition, we provide a systematic approach that allows conflict minimization during SADP layout decomposition. Our main contributions include:

- We present a new layout decomposition framework for SADP and complementary EBL, which considers overlay minimization and EBL throughput optimization simultaneously.
- We propose a new graph formulation and a matching-based approach that allows eliminating conflicts by the merge-and-cut technique.
- We show that for pure SADP layout decomposition problem, our approach can be adapted to minimize conflicts with overlay consideration.
- The results show that our approach is very efficient, and that all conflicts can be eliminated with minimal overlay error and e-beam utilization.

The rest of Section 2.5 is organized as follows. We will introduce the merge-and-cut technique and give the problem formulation for hybrid lithography in Section 2.5.1. In Section 2.5.2, we present our *face graph* formulation that embeds SADP constraints as well as the solution candidates for solving

conflicts. Our layout decomposition that performs simultaneous overlay and EBL throughput optimization is presented in Section 2.5.3. We then explain an adapted conflict minimization approach that can be used for pure SADP in Section 2.5.4. Finally, we will show experimental results in Section 2.5.5, followed by the summary in Section 2.5.6. The preliminary results of this work were reported at [34].

2.5.1 Merge-and-cut Technique

The layout decomposition problem is usually formulated as a two-coloring problem, where conflicting patterns must be assigned different colors. One color will be defined by mandrel pattens, as pattern A in Fig. 2.1, while the other will be defined by non-mandrel patterns, as B. A two-coloring result of Fig. 2.23(a) is shown in 2(b).

The challenge of SADP layout decomposition is that two-coloring method may not necessarily avoid all conflicts. To further eliminate conflicts, *merge-and-cut* technique is utilized [9,58] to merge two conflicting patterns and then trim out the unwanted part by the trim mask. Fig. 2.23(b) shows two conflicts remaining after two-coloring, and thus we cannot generate those patterns by the mandrel and trim mask directly. Fig. 2.23(c)$\sim$(e) show possible merge-and-cut solutions by merging two conflicting patterns and then cutting the unwanted area by the *cutting patterns* $cut_1 \sim cut_6$ defined by the trim mask. As mentioned previously, the pattern boundaries that directly touch the trim patterns would have potential overlay error; meanwhile, cut-

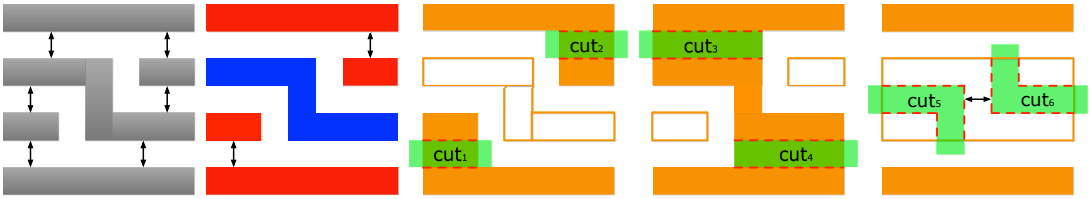


Figure 2.23: Merge-and-cut example. (a) Target layout. (b) Two- coloring result. (c)∼(e) Layout decomposition with merge-and-cut. Red lines show boundaries with overlay error risk.

ting patterns cannot violate the minimum spacing S_{dp} such as (e). Therefore, merge-and-cut solutions should be selected appropriately such that the cutting boundaries/overlay are as small as possible. It can be seen that the solution in Fig. 2.23(c) cause shorter boundaries with overlay risk than that of (d).

2.5.1.1 Problem Formulation for Hybrid SADP and EBL

Given a layout with 2D random patterns, our task is to perform layout decomposition with hybrid SADP and EBL. Because there may not be a conflict-free solution with pure SADP, we utilize EBL as an extra cutting lithography. Our objective is to remove all conflicts by the merge-and-cut technique, while minimize the overlay error and the required e-beam cost (defined in Section 2.5.3).

2.5.2 Graph Formulation with Embedded SADP Constraints and Conflict Solving Solutions

We first introduce our graph formulation that is essential for our layout decomposition approach. There are three purposes of this graph:

1. To embed the minimum S_{dp} constraint.

2. To generate all solution candidates for solving conflicts.

3. To formulate the cost of the layout decomposition.

Take Fig. 2.25 as an example. There are two conflicts after performing two-coloring for the target layout as shown in (a), and two possible solutions are shown in (b), (c). Several techniques needs to be applied to embed the SADP constraint and solution candidates for solving conflicts in the face graph. In the following, we will introduce these techniques which are applied as shown in the flow in Fig. 2.24.

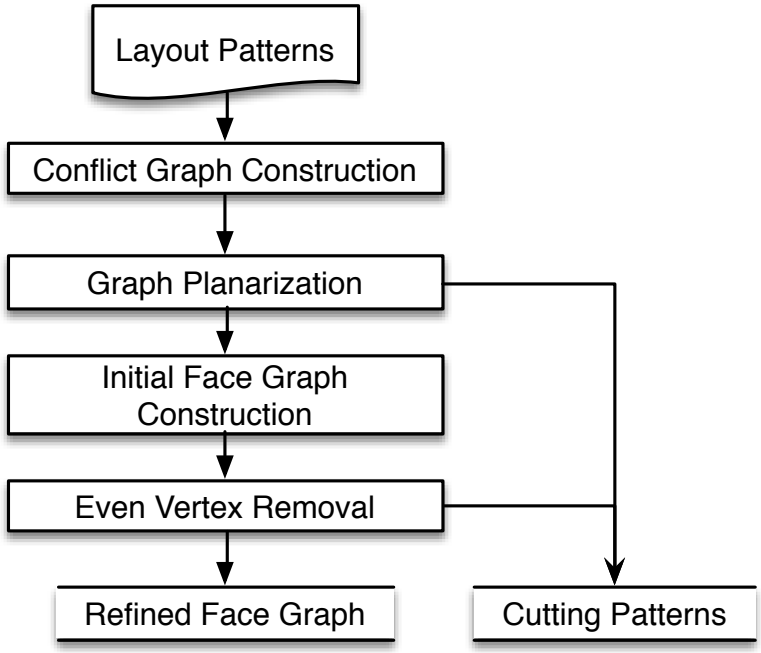


Figure 2.24: Face graph construction flow.

2.5.2.1 Conflict Graph and Face Graph Construction

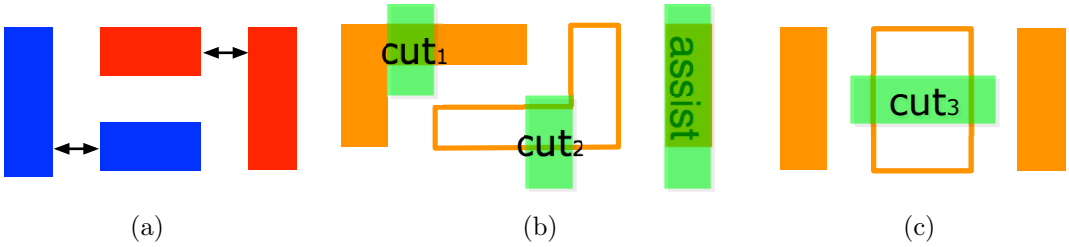


Figure 2.25: (a) Target layout. (b)(c) Merge-and-cut solutions.

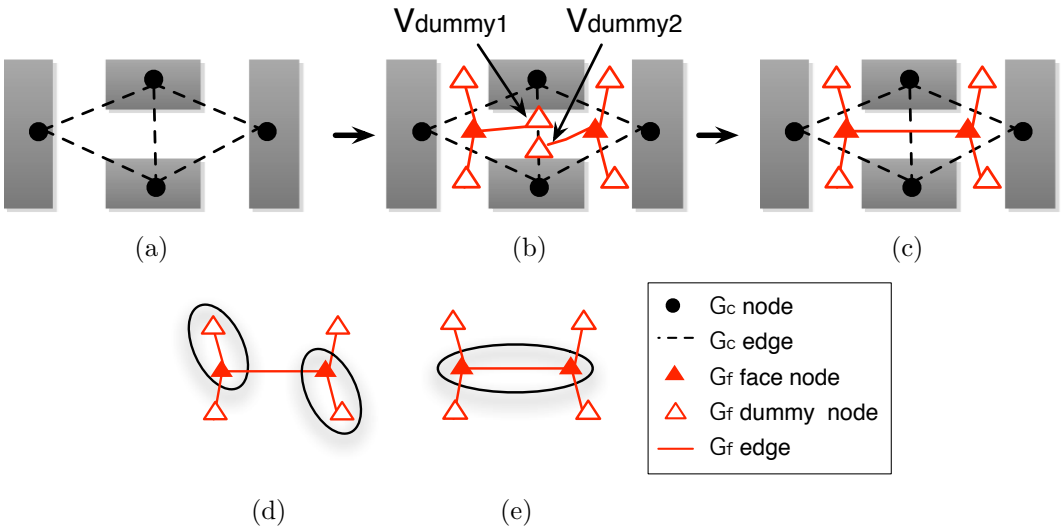


Figure 2.26: (a)(b)(c) Conflict graph (black) and face graph (red) example. (d)(e) Matching results.

Given a 2D layout, we construct a conflict graph $G_c = (V_c, E_c)$ to express the relationship among layout patterns. Each vertex $v_c \in V_c$ represents a pattern, and each edge $e_c \in E_c$ is constructed when the distance between two patterns is less than S_{dp}. Fig. 2.26(a) show the conflict graph of the layout in Fig. 2.25(a). It has been shown that in two-coloring problem, a conflict

occurs only when there is an odd cycle in the conflict graph [45]. To achieve a valid SADP layout decomposition, we have to eliminate all odd cycles in G_c.

Based on G_c, we define its dual graph, face graph $G_f = (V_f, E_f)$ where $V_f = V_{face} \cup V_{dummy}$. A vertex $v_{face} \in V_{face}$ corresponds to a face in G_c except the exterior face, and a *dummy vertex* v_{dummy} is created for each merge-and-cut candidate of a conflict. An edge $e_f \in E_f$ connects v_{face} and v_{dummy} if v_{dummy} is the solution candidate to solve the corresponding conflict of v_{face}. We call v_{face} as an *even (odd) vertex* if it corresponds to an even (odd) face in G_c. The initial face graph of the layout in Fig. 2.25(a) is shown in red in Fig. 2.26(b).

For adjacent odd vertices v_{f1} and v_{f2}, they may share the same merge-and-cut candidates. For example, v_{dummy1} and v_{dummy2} in Fig. 2.26(b) both refer to cut_3 in Fig. 2.25(c). In the case where $e_{f1} = (v_{f1}, v_{dummy1})$ and $e_{f2} = (v_{f2}, v_{dummy2})$ refer to the same merge-and-cut candidate, we combine e_{f1} and e_{f2}, and remove the dummy vertices v_{dummy1} and v_{dummy2}, as shown in Fig. 2.26(c). As shown in Fig. 2.26(c), the two dummy vertices in the middle are removed.

2.5.2.2 Conflict Graph Partitioning

Decompsing all layout patterns at the same time may consume a lot of computational time. In fact, in most cases, patterns can be partitioned into smaller groups such that divide-and-conquer approach can be applied to solve them. If there is no edges between two vertices v_i and v_j in G_c, then v_i and v_j

can be solved independently. This is straightforward since the coloring result of v_i does not affect the coloring of v_j when their distance is larger than S_{dp}.

Once we construct G_c, we traverse it from an arbitrary vertex by DFS to obtain a connected component. Then we pick an untraversed vertex as a new head of DFS to find another connected component until all vertices are traversed. For each connected component, we construct a sub-G_f based on the corresponding sub-G_c. The following layout decomposition procedures will be performed for each sub-G_f individually.

2.5.2.3 Conflict Graph Planarization

It has been shown in [27, 77] that the planarity of the conflict graph is based on the setting of S_{dp}. The conflict graph is planar only if Eq. (2.5) is satisfied:

$$\begin{cases} S_{dp} < 2 \times S_{min} & \text{in the Manhattan distance} \\ S_{dp} < \sqrt{2} \times S_{min} & \text{in the Euclidean distance} \end{cases} \tag{2.5}$$

, where S_{min} is the minimum spacing between patterns on the layout. In the case that Eq. (2.5) is violated, we need to planarize G_c since G_f cannot be constructed based on a non-planar graph.

If a conflict graph G_c is highly non-planar, it implies that several patterns conflict with multiple patterns. Therefore it is less possible to find a valid DPL decomposition. Be assuming that the non-planar cases in the give layout is limited, we apply the following heuristic to solve the non-planar subgraph. In a non-planar graph G_c, assume $e_1 \in E_c$ and $e_2 \in E_c$ cross each

other, we eliminate one of the two edges by merging their connected vertices. Conceptually, this planarization means we force two patterns to be merged to prevent a non-planar case. In order to minimize the overlay error and EBL cost of merging two patterns, the edge with smaller merging cost (defined in Section 2.5.3.2) will be eliminated.

2.5.2.4 Even Vertex Removal for Face Graph

The *degree* of a face is defined by the number of edges that bound the face. Given an edge $e_f = (v_{face}, v_{dummy}) \in G_f$, it implies that we can use the merge-and-cut candidate e_f to reduce the degree of the corresponding face of v_{face} by one. Since V_f contains vertices from all faces, our merge-and-cut candidate may either make an odd face become an even face (meaning the conflict is solved), or make an even face become an odd face (meaning a new conflict is introduced). Because applying merge-and-cut increases the risk of overlay error on the cutting boundaries, we would like to minimize new conflicts introduced by merge-and-cut. With this motivation, we apply a vertex removal heuristic in Alg. 3 to greedily remove even vertices in G_f.

We prefer not to introduce new conflicts in Line 6, and apply greedy merging in Line 8 if introducing conflicts is not preventable. Note that Line 8 will introduce a new odd vertex by combining an odd face with an even face. In our implementation, we use a queue to store all initial odd vertices and and keep updating newly introduced odd vertices with the above check until the queue is empty.

2.5.3 Layout Decomposition with SADP and Complementary EBL

In complex designs, the layout patterns may not be able to be manufactured simply by SADP with merge-and-cut techniques. Hybrid lithography techniques can increase the layout flexibility and further push the resolution limit.

2.5.3.1 SADP with Complementary E-beam Lithography

A limitation of applying the merge-and-cut technique with the trim mask is that the distance between cutting patterns may violate the minimum DPL spacing S_{dp}. For example, the solution in Fig. 2.23(e) requires two cutting patterns cut_5 and cut_6, which actually conflict with each other. Motivated by this limitation, we incorporate complementary EBL into the conflict elimination process. Since EBL enables higher pattern resolution, the merge-and-cut technique would be less restricted than S_{dp}.

Algorithm 3 RemoveEvenVertex

```
1: F_odd ← all odd faces in G_c
2: for all f ∈ F_odd do
3:    Γ(f)_even ← even adjacent faces of f
4:    Γ(f)_odd ← odd adjacent faces of f
5:    if Γ(f)_even ≠ φ then
6:       Remove v_i ∈ G_f, ∀f_i ∈ Γ(f)_even
         where v_i is the corresponding face vertex of f_i
7:    else
8:       Solve f by the min-cost merge-and-cut candidate as Section 2.5.3.2
9:    end if
10: end for
```

EBL enables smaller feature width and spacing, and thus it achieve better pattern quality and design flexibility than SADP. However, EBL throughput is its biggest bottleneck as the write time is determined by the number of e-beam shots. Therefore, extensive use of e-beam cutting is not practical for manufacturing. With hybrid SADP and EBL process, we should generate a manufacturable SADP layout with minimal overlay error, and meanwhile, utilize minimal e-beams to solve the remaining conflicts.

We adopt the conventional e-beam system where e-beam shots are variable-shaped (rectangular) beams (VSB). Cutting patterns formed by VSB require layout fracturing, meaning the patterns are decomposed into non-overlapping e-beam shots/rectangulars. For example, each of $cut_1 \sim cut_4$ in Fig. 2.23 requires one VSB, while each L-shape of cut_5 and cut_6 requires two VSBs.

2.5.3.2 Min-Cost Matching based Conflict Elimination

In SADP layout decomposition problem, our objective is to eliminate conflicts with minimal overlay error and EBL utilization. We first explain our min-cost matching based conflict elimination algorithm on the *face graph*. Then we discuss how to utilize this algorithm for the overlay and EBL co-optimization, including a post processing based approach (Section 2.5.3.2) and a simultaneous optimization (Section 2.5.3.2). Two layout decomposition approaches are proposed, a post processing based approach (Section 2.5.3.2) and a simultaneous optimization (Section 2.5.3.2).

The face graph defined in Section 2.5.2 has the following property:

PROPERTY A. An edge $e_f = (v_{odd}, v_{dummy}) \in E_f$ maps to a merge-and-cut candidate of the corresponding conflict of v_{odd}, where merge-and-cut reduces the degree of v_{odd} by one.

According to Property A, we can solve a conflict corresponding to an odd vertex by selecting one of its connecting edges $e_f \in E_f$. It is obvious that selecting more than one e_f for an odd vertex is unnecessary because a new conflict would be introduced and more merge-and-cut efforts would be required. Therefore, we seek to find one merge-and-cut solution for each conflict, and we formulate the conflict elimination problem as the matching problem, where each match corresponds to solving a conflict. For example, Fig. 2.26(d) and (e) show two different matching results which corresponds to the final masks shown in Fig. 2.25(b) and (c), respectively.

Post Processing Based Conflict Elimination Since the trim mask itself can be used for the merge-and-cut technique to solve conflicts, we can view EBL cutting as a back-up solution during conflict elimination. We propose a two-stage approach for overlay error and e-beam optimization. First, we solve all conflicts by applying the min-cost matching algorithm on G_f, then we assign the obtained cutting patterns to the trim mask and e-beam shots according to SADP constraint S_{dp}. The approach flow is shown in Alg. 4, where $N_{shot}(e)$ represent the required number of e-beam shots for the merge-and-cut candidate corresponds to e.

Algorithm 4 Hybrid-Post

Input: $G_f = (V_f, E_f)$ // Section 2.5.2

Output: $P_{cut} = P_{optical} \cup P_{ebl}$, with the objective of minimizing $\sum_{e_i \in P_{optical}} cost_{e_i}$ and $\sum_{e_j \in P_{ebl}} N_{shot}(e_j)$

1: AssignCost_SADP(E_f)
2: $P_{allcuts} \leftarrow$ RunMatching(G_f)
3: $G_{mc} \leftarrow$ ConstructConflictGraph($P_{allcuts}$)
4: $P_{optical} \leftarrow MIS(G_{mc})$
5: $P_{ebl} \leftarrow P_{allcuts} - P_{optical}$

In the beginning of Alg. 4, G_f is constructed based on Section 2.5.2. Since EBL is not considered in the first stage, we model the edge cost simply by the SADP overlay error according to Eq. (2.6) in Line 1.

$$cost_e = L_{boundary}(e) \quad \forall e \in E_f \tag{2.6}$$

where $L_{boundary}$ is the boundary length of the corresponding cutting pattern of e.

We then solve all conflicts by merge-and cut technique (Line 2). The cutting patterns $P_{allcuts}$ with the minimum overlay can be obtained by performing the min-cost matching algorithm in Line 2. However, there may exist conflicts among the cutting patterns because of the S_{dp} constraint. With e-beam available, we can carefully select a subset of cutting patterns $P_{optical}$ that do not violate S_{dp}, and let the rest of the cutting patterns P_{ebl} formed by EBL. We first construct a conflict graph G_{mc} for $P_{allcuts}$ in Line 3 to check if there is any conflict among $P_{allcuts}$. In order to minimize e-beam shot utilization, we apply maximal independent set (MIS) algorithm on G_{mc} to obtain the maximal number of valid patterns for $P_{optical}$ (Line 4), and assign the rest of the

cutting patterns to be done by EBL (Line 5). Based on the property of MIS, the S_{dp} constraint is guaranteed to be satisfied for $P_{optical}$.

Note that the S_{dp} constraint is guaranteed to be satisfied for $P_{optical}$ according to Theorem 2.

Theorem 2: *There is no conflict for patterns in $P_{optical}$ obtained in Alg. 4.*

PROOF In the conflict graph G_{mc}, a conflict edge connects two vertices if their corresponding patterns cannot co-exist on the trim mask. For any two vertices in the MIS of $P_{optical}$, there must be no edge connecting them according to the definition of independent set. Therefore, no conflict exists for patterns in $P_{optical}$

Conflict Elimination with Simultaneous Overlay and EBL Throughput Optimization Although the approach in Section 2.5.3.2 can successfully solve conflicts with hybrid SADP and EBL, it only considers EBL in the last stage and does not include e-beam optimization when finding the min-cost matching. To further improve the decomposition result, we propose a simultaneous overlay error and EBL throughput optimization as shown in Alg. 5. The main idea of the algorithm is to start from a restricted solution space and gradually increase the solution space with more EBL merge-and-cut candidates until we find a valid solution. Based on the algorithm, only necessary e-beam candidates are considered, and the matching algorithm can simultaneously optimize SADP overlay error and e-beam utilization.

Algorithm 5 Hybrid-Sim

Input: $G_f = (V_f, E_f)$ // Section 2.5.2

Output: $P_{cut} = P_{optical} \cup P_{ebl}$, with the objective of minimizing $\sum_{e_i \in P_{optical}} cost_{e_i}$ and $\sum_{e_j \in P_{ebl}} N_{shot}(e_j)$

1: AssignCost_SADP(E_f)
2: $P_{conf} = \Phi$
3: **repeat**
4: $P_{optical}, P_{ebl} \leftarrow$ RunMatching(G_f)
5: $P_{conf} \leftarrow$ ValidateCut($P_{optical}$)
6: SubstituteEBL(P_{conf})
7: **until** $P_{conf} == \Phi$

In the beginning of Alg. 5, G_f is constructed based on Section 2.5.2. All edges are initialized as optical cuts with the cost defined by Eq. (2.6). We then iteratively perform min-cost matching algorithm in Line 4 to find the cutting patterns. Since we may obtain conflicts among some optical cutting patterns in Line 5, we substitute those conflicting optical cuts by EBL cuts in Line 6 with the cost function in Eq. (2.7).

$$cost_e = C_{ebl} \times N_{shot}(e) \quad \forall e \in P_{conf} \tag{2.7}$$

where C_{ebl} is a user-defined parameter to control the cost of a e-beam shot.

In our implementation, we set C_{ebl} sufficiently large than the cost of any optical cut, such that optical cuts are always preferred than EBL cuts. However, C_{ebl} can also be properly defined to trade e-beam shots for overlay error improvement. By including both overlay and e-beam cost with Eq. (2.6) and Eq. (2.7), our min-cost matching solution can minimize the overlay error and e-beam utilization simultaneously. Note that we assume rectangular beam

shape is applied. For example, an L-shaped pattern requires at least two beam shots. In addition, there are minimum size and maximum size limitation for beam shape, which would also affect the number of shots N_{shot} of each merge-and-cut candidate. Because a merge-and-cut candidate is formed between patterns with half-pitch width, generally the minimum shape constraint would not be violated; and the maximum shape constraint would apply for long cutting patterns. Fig. 2.27 shows the matching solutions of Alg. 5 and how cutting candidates are updated in each iteration.

Our min-cost matching in Line 4 is implemented by LEDA library with the complexity $O(V_f E_f \lg V_f)$. The validation and substitution in Line 5-6 can be done by querying our pre-constructed conflict graph for all merge-and-cut candidates, which takes near linear time. Although Alg. 5 needs to perform the above processes repeatedly, the iterations would converge very quickly because the solution space is enlarged in each iteration. In our experiment, all cases are finished in less then ten iterations.

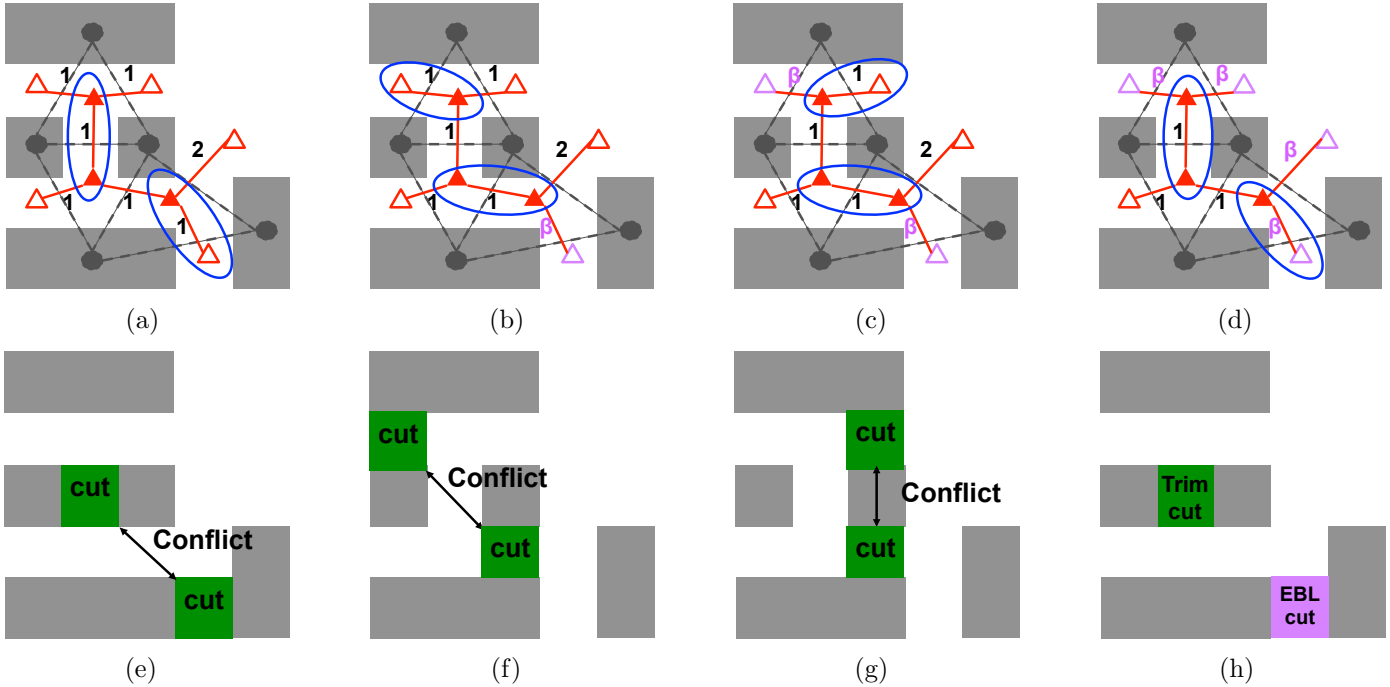


Figure 2.27: Example of Hybrid-sim algorithm. The edge cost shows the overlay penalty of a cut, where β indicate the cost of one e-beam shot.(a)∼(d) Matching solutions obtained in each iteration; cut candidates are replaced as e-beam if there is a conflict in the matching solution. (e)∼(h) The corresponding merge-and-cut solutions of (a)∼(d). The algorithm continues until there is no conflict as shown in (h).

2.5.4 Overlay-aware Conflict Minimization for Pure SADP

The approaches discussed so far are targeted at the layout decomposition for hybrid SADP and EBL. We find that our approach can be adapted for conflict minimization in pure SADP layout decomposition. Since there is no previous study that optimize cutting patterns in SADP layout decomposition, this approach can be very useful when the layout is highly complicated and complementary lithography is not available. We first introduce the problem formulation, and explain how to adapt our face graph to solve this problem.

2.5.4.1 Problem Formulation

Given a layout with 2D random patterns, our task in overlay-aware conflict minimization is to perform SADP layout decomposition with the manufacturing constraints on the mandrel and trim masks. Our objective is to solve as many conflicts as possible by merge-and-cut technique, while minimize the overlay error introduced by the cutting patterns.

2.5.4.2 Adapted Face Graph for Conflicting Cuts

In the face graph defined in Section 2.5.2, the merge-and-cut candidates may conflict each other. Therefore, when applying the min-cost matching algorithm to solve conflicts as explained in Section 2.5.3.2, we may obtain conflicting cutting patterns. These conflicts must be prevented in pure SADP layout decomposition.

Fig. 2.28(a) shows the layout patterns and its corresponding conflict

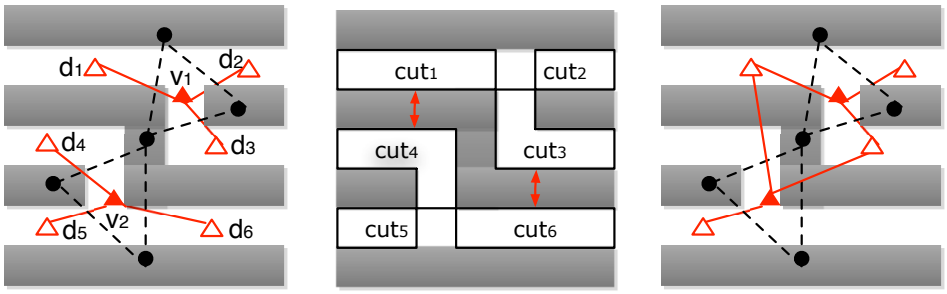


Figure 2.28: Face graph adapted for conflicting cuts. (a) Conflict graph (black) and Face graph (red). (b) Cutting patterns that cannot co-exist are indicated by arrows. (c) Adapted face graph without conflicting cuts.

and face graph. Two conflicts v_1 and v_2 are discovered because they forms odd cycles in the conflict graph. The merge-and-cut candidates of v_1 and v_2 are shown by $cut_1 \sim cut_3$ and $cut_4 \sim cut_6$ in Fig. 2.28(b), respectively. It can be seen that if we select cut_1 and cut_4 to solve the two conflicts, the solution would not be valid because the two cuts are too close to be fabricated on the same mask. Similarly, cut_3 and cut_6 cannot co-exist.

To add the conflicting cuts information into our face graph, we check all merge-and-cut candidates by traversing all edges, and make conflicting edges connect to the same dummy vertex. Finally, we can obtain an adapted face graph G'_f with the conflicting cuts information embedded. As shown in Fig. 2.28(c), after this graph adaption, d_1 and d_4 is merged, and so is d_3 and d_6.

2.5.4.3 Matching based Conflict Minimization

The adapted face graph G'_f has the following property:

PROPERTY B. If two merge-and-cut candidates cannot co-exist because their spacing is less then S_{dp}, the corresponding edges e_{f1} and e_{f2} connect to the same $v_{dummy} \in V_{dummy}$.

Property B guarantees that the matching algorithm would only select one edge that covers the same dummy vertex. This ensures that S_{dp} rule is satisfied for the cutting patterns. Besides, Property A in Section 2.5.3.2 still holds for the adapted face graph. Consequently, by modeling the edge cost of G'_f with the desired SADP cost, we can minimize conflicts with the min-cost matching algorithm presented in Section 2.5.3.2. Here our objective is to minimize overlay error introduced by cutting patterns, therefore, Eq. (2.6) is adopted in the matching problem. This approach allows simultaneous optimization for overlay minimization and conflict minimization for pure SADP layout decomposition.

2.5.5 Experimental Results

The proposed algorithms are implemented in C++ and tested on Intel platform with 2.66 GHz CPU and 4G memory. We synthesize OpenSPARC T1 designs with Nangate 45nm standard cell library [5], and perform placement and routing with Cadence SOC Encounter [1] to generate the layouts. These layouts are then scaled down for 22nm technology node. For simplicity, we assume the sizes of the minimum pattern width, spacing, and spacer width are 50nm, and make the corresponding adjustment for the benchmark. The double patterning spacing S_{dp} is set large enough to introduce conflicts to

evaluate the performance of our algorithms. Table 2.7 shows the statistics of five designs with the number of 2D patterns (#Polygon) and the initial coloring conflicts ($\#Conf$) before applying our approaches, where $\#Conf$ is obtained based on the two-coloring result.

We first present the layout decomposition results for pure SADP lithography based on the approach in Section 2.5.4. This shows the effectiveness of our conflict elimination approach using cutting patterns formed by the trim mask, but also shows the limitation of pure SADP. Then we show how layout decomposition with hybrid SADP and EBL can further improve the results.

Table 2.7: Overlay-aware layout decomposition for SADP.

Design	#Polygon	$\#Conf$	SADP-Cut			SADP-OV			
			$\#Conf_{rem}$	$Bndy_{ov}$ (um)	CPU (s)	$\#Conf_{rem}$	$Bndy_{ov}$ (um)	CPU (s)	OV Imp%
alu	9792	1992	46	541.98	1.57	60	311.79	1.64	42.47%
byp	27675	5015	274	1413.04	2.56	25	1129.90	2.83	20.04%
div	20501	3914	222	901.16	3.32	39	680.20	3.36	24.52%
ecc	7922	2282	104	575.29	1.36	24	430.48	1.42	25.17%
efc	7173	1988	91	503.44	1.15	96	354.47	1.14	29.59%
Average									28.36%

2.5.5.1 Overlay-aware Layout Decomposition for SADP

We first apply the proposed approach in Section 2.5.4 to solve conflicts with the trim mask for conflict and overlay error minimization. Because the existing approaches for SADP layout decompositions [9, 61, 75, 90] are performed for two-colorable cases, no solution would be generated for comparison on designs with conflicts. Although layout perturbation [35] can be applied to solve native conflicts, we do not allow layout change in our problem. An alternative to solve this problem is to minimize the total number of cutting patterns, by which we expect less overlay error and unsolved conflicts because less cutting patterns compete for the mask resource. As a baseline, we implement this alternative (SADP-Cut) by replacing Eq. (2.6) with Eq. (2.8) and compare it with the proposed overlay-aware layout decomposition (SADP-OV).

$$cost_e = 1 \quad \forall e \in E_f \tag{2.8}$$

Table 2.7 shows the results after layout decomposition in terms of the remaining conflicts ($\#Conf_{rem}$), the total length of the overlay-risky boundaries touched by the trim mask ($Bndy_{ov}$), and the CPU time. It can be seen that the merge-and-cut technique is not sufficient to solve all conflicts because of the resolution limit of the trim mask. Compared with the baseline, SADP-OV successfully reduces the overlay error, whose effect can be represented by the length of the cutting boundaries. On average, the overlay error can be reduced by 28.36% with SADP-OV. Although there are still outstanding conflicts that cannot be resolved, our approach successfully resolve more

than 95% of the initial conflicts, showing that merge-and-cut is promising for SADP layout decomposition. Note that our benchmarks are not targeting at any specific lithography process and thus are not SADP-friendly designs. By properly designing the layout for SADP or specify lithography-aware rules in early design stages, it would be easier to solve conflicts by our approach.

It is interesting to observe that in some cases (byp, div and ecc), SADP-OV reduces conflicts more effectively than SADP-Cut. We further investigate these cases and found that the merge-and-cut candidates selected by SADP-OV are often shorter and simpler (more rectangles rather than L-shape or Z-shape) than those by SADP-Cut. The side effect is due to that the cutting patterns from SADP-OV are less likely to conflict with each other. Consequently, more cutting patterns are valid and more conflicts can be solved.

2.5.5.2 Overlay and EBL Throughput Co-optimization for SADP with Complementary EBL

By adopting complementary EBL, the conflicts that cannot be handled in Section 2.5.5.1 can be solved. We compare the layout decomposition results when applying the two conflict elimination approaches, Hybrid-Post and Hybrid-Sim. Because Hybrid-Post is a two-stage approach, we would like to study how the result in the first stage affects the final result. Therefore, two versions of Hybrid-Post are implemented, one perform the min-cost matching algorithm based on Eq. (2.6) (Hybrid-Post-OV), while the other perform the min-cost matching algorithm based on Eq. (2.8) (Hybrid-Post-Cut).

The results are shown in Table 2.8, where $\#VSB$ refers to the total number of variable shaped beams. Note that the cutting patterns are 2-dimensional and thus a conflict may require more than one VSB to solve. With complementary EBL, all conflicts in our benchmark are solved. It can be seen that Hybrid-Post-Cut requires a large number of VSB because it leaves more unsolved conflicts before applying e-beams. The simultaneous optimization Hybrid-Sim outperforms the two post-processing based approaches, which reduces VSB utilization by 69% while achieving comparable overlay error minimization with Hybrid-Post-OV.

Applying Hybrid-Post does not increase much computational time compared to Table 2.7. Although Hybrid-Sim iteratively performs matching algorithm, the iterations converge quite fast and thus does not cause much runtime overhead. In our experiment, at most 4 iterations are needed to obtain a valid layout decomposition solution.

Table 2.8: Layout decomposition for Overlay and EBL throughput co-optimization

Design	Hybrid-Post-Cut			Hybrid-Post-OV			Hybrid-Sim		
	#VSB	$Bndy_{ov}$ (um)	CPU (s)	#VSB	$Bndy_{ov}$ (um)	CPU (s)	#VSB	$Bndy_{ov}$ (um)	CPU (s)
alu	240	541.98	1.59	219	311.79	1.66	24	329.00	1.85
byp	1347	1413.04	2.63	60	1129.90	2.91	11	1133.85	3.36
div	1249	901.16	3.37	119	680.20	3.41	72	685.81	3.77
ecc	479	575.29	1.38	72	430.48	1.45	37	433.41	1.52
efc	561	503.44	1.16	335	354.47	1.16	54	378.88	1.36
Avg Ratio	8.47	1.41	0.96	1.00	1.00	1.00	0.31	1.03	1.12

2.5.6 Summary

We present a new layout decomposition framework for SADP and complementary EBL, which considers overlay minimization and EBL throughput optimization simultaneously. We show that conflict elimination by merge-and-cut can be formulated as a matching-based algorithm based on our graph formulation. Our approach is flexible to be applied for different lithography resources, including SADP with complementary EBL and pure SADP. The results show that applying merge-and-cut technique in hybrid SADP and EBL layout decomposition is promising, and that our approaches is efficient and effective in minimizing overlay error and e-beam utilization simultaneously.

Chapter 3

Mask Optimization With Process Window Aware Inverse Correction

As technology nodes continue shrinking, semiconductor industry is still stuck at 193nm lithography. Due to the resolution limit, various resolution enhancement techniques (RETs) have been proposed to achieve deep sub-wavelength lithography. Optical Proximity Correction (OPC) is one of the RETs that have been widely used.

Typical OPC approaches can be divided into two categories: rule-based approach [71] and forward model-based approach [16]. Rule-based OPC is simple and fast, but only suitable for less aggressive designs. Forward model-based OPC usually relies on edge fragmentation and movement, where mask is adjusted iteratively based on mathematical models. To allow more flexibility, a topological invariant pixel based OPC [83] was proposed. However, the solution space of these approaches is natively limited and thus OPC in advance technology nodes has become more challenging. Inverse models-based method, also referred as Inverse lithography technique (ILT) [36,65], is one of the strong OPC candidates for 32nm and beyond [73].

ILT-based OPC solves the inverse problem of the imaging system through

optimizing an objective function. The ILT process starts from the target printed patterns and iteratively optimizes the mask. ILT approaches are expected to achieve better results than conventional OPC methods because its pixelated mask optimization enables better contour fidelity.

In recent years, ILT has drawn more attention because of its great flexibility in mask optimization. Granik [37] proposed a fast solution based on constrained nonlinear formulation. Shen et al. [68] formulated ILT as a nonlinear image restoration problem, and solved it by a level-set time-dependent model. Poonawala et al. [66] formulated the inverse problem as a continuous function and optimized the mask by the gradient descent approach. Various enhancement techniques [42, 55, 93] have been presented based on the gradient descent framework. Zhang et al. proposed cost function reduction methods [91, 92] to make the optimization less dependent on the initial condition. However, most of these approaches only optimized image contour, and only [42] considered the focus variation. Moreover, none of them can directly optimize edge placement error (EPE), which is an important measurement for yield impact.

The main objective for OPC is to obtain an optimized mask that can compensate the pattern distortion. However, as the feature size is getting smaller, the yield impact of layout uncertainty during the manufacturing process is getting larger. Considering manufacturing variability has thus become an important issue for mask optimization and has been studied in several forward model-based OPC methods [10, 49, 85]. In order to tackle the above issues in ILT, in this chapter, we propose new mask optimization approaches

considering simultaneously 1) the design target optimization under nominal process condition and 2) process window minimization with different process corners. The main contributions include:

- We propose mask optimizing approaches considering design target and process window simultaneously.
- We formulate the EPE violation as a sigmoid function and derive the closed form of its gradient for EPE minimization.
- We present MOSAIC˙exact that achieves the best results among all compared approaches, and MOSAIC˙fast with efficient gradient computation.
- We perform experiments on 32nm M1 designs released by IBM and show that our two approaches outperform the first place winner of the ICCAD 2013 contest by 7% and 11%, respectively.

The rest of the chapter is organized as follows. We first give an introduction of the forward lithography process in Section 3.1. Our mask optimization approaches are explained in Section 3.2. Finally, we show our experimental results and comparison in Section 3.3, followed by the summary in Section 3.4. The preliminary results of this work were reported at [31].

Table 3.1: Variable and symbol definitions.

Variables	Definitions
M	Mask
I	Intensity after optical system
Z	Printed pattern after photoresist process
N	Length/Width of the mask
$\otimes$	Convolution operator
$\odot$	Element-by-element multiplication

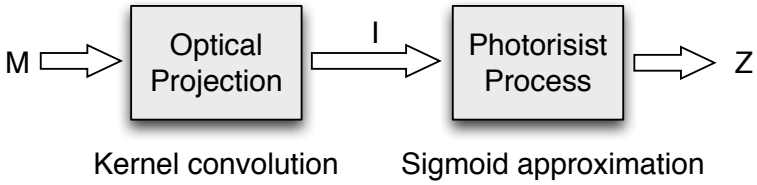


Figure 3.1: Forward lithography process model.

3.1 Forward Lithography

We first explain the mathematical form of the forward lithograph process. Table 3.1 gives the basic variables and operators. The lithography process is shown as Fig. 3.1. The mask M is projected through optical lens onto the wafer plane, which is coated with photoresist. The aerial image I then goes through development and etching processes to form the final printed image Z. The forward lithography process of obtaining printed image from a given mask can be modeled with two phases, optical projection model and photoresist model.

The Hopskins diffraction model [40] has been widely used for partially coherent imaging system. To reduce the computational complexity, we adopt

the singular value decomposition model (SVD) [15] to approximate the Hopskins model in this section. In SVD model, the Hopskins diffraction model can be decomposed into a sum of coherent systems based on eigenvalue decomposition as Eq. (3.1).

$$I(x,y) = \sum_{k=1}^{N^2} w_k |M(x,y) \otimes h_k(x,y)|^2, \quad x,y = 1,2,...N \tag{3.1}$$

where h_k is the kth kernel of the model and w_k is the corresponding weight of the coherent system. The N_hth order approximation to the partially coherent system can be obtained by

$$I(x,y) \approx \sum_{k=1}^{N_h} w_k |M(x,y) \otimes h_k(x,y)|^2. \tag{3.2}$$

In our implementation, the system is approximated with $N_h = 24$ kernels.

The light transmitted through the mask is then exposed on the photoresist. An image can be developed if the light intensity of the exposed area exceeds a threshold th_r. Therefore, the photoresist effect can be defined by the following step function:

$$Z(x,y) = \begin{cases} 0 & \text{if } I(x,y) \leqslant th_r \\ 1 & \text{if } I(x,y) > th_r \end{cases} \tag{3.3}$$

Later in this chapter, we will derive the partial differential of the imaging system. In order to obtain a continuous form, we apply the sigmoid function to approximate the threshold model:

$$Z(x,y) = sig(I(x,y)) = \frac{1}{1+e^{-\theta_Z (I(x,y) - th_r)}} \tag{3.4}$$

where θ_Z defines the steepness of the sigmoid function. Fig. 3.2 illustrates our sigmoid function with $\theta_Z = 50$ and $th_r = 0.225$.

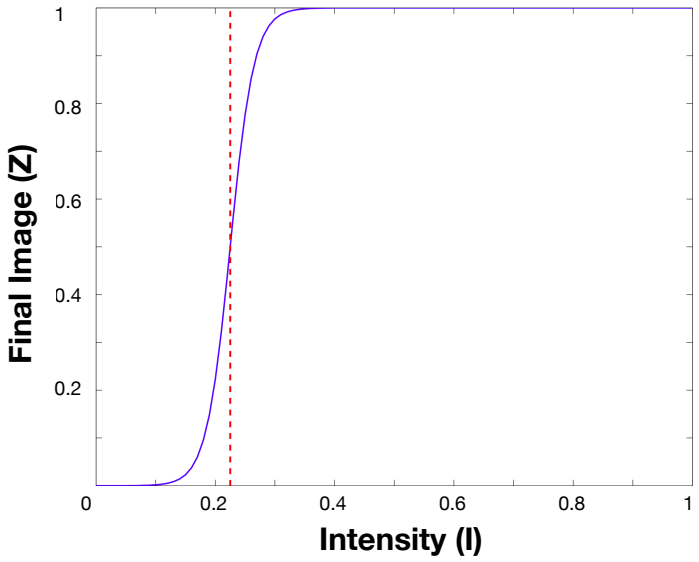


Figure 3.2: Sigmoid function with $\theta_Z = 50$ and $th_r = 0.225$.

3.2 Mask Optimization for Design Target and Process Window

3.2.1 Inverse Lithography based on Gradient Descent

The forward lithography process in Eq. (3.4) can be described below:

$$Z = f(M) \tag{3.5}$$

The OPC problem by inverse lithography tries to find:

$$M_{opt} = f^{-1}(Z_t) \tag{3.6}$$

where Z_t is the target pattern and M_{opt} is the optimized mask with OPC. However, this is an ill-posed problem because different masks may yield the same result. Therefore, there is no directed closed form solution to Eq. (3.6).

Algorithm 6

1: $F \leftarrow$ objective function of OPC
2: $M \leftarrow Z_t$ with rule-based SRAF
3: $P \leftarrow$ initialize unconstrained variables corresponding to M
4: **repeat**
5: $\quad \mathbf{g} \leftarrow$ calculate gradient ∇F
6: $\quad P \leftarrow P - stepSize \times \mathbf{g}$
7: $\quad M \leftarrow$ recalculate pixel value based on P
8: **until** $\#iteration = th_{iter}$ or $RMS(g)$ ¡ th_g
9: $M_{opt} \leftarrow M^{iter}$ with the lowest objective value

Instead, gradient descent based approaches have been commonly used to solve the ILT problem.

The details of our methodologies to solve the ILT problem are shown in Alg. 6. The ILT problem is formulated as a multivariable objective function F where each variable $p(x, y) \in P$ corresponds to a pixel of the mask. As explained previously, our objective in this work is to optimize the design target and the process window, represented and evaluated below.

$$\begin{array}{ll} \text{Minimize:} & F = \alpha \times \#EPE\ Violation + \beta \times PV\ Band \\ \text{Subject to:} & M(x, y) \in \{0, 1\} \end{array} \tag{3.7}$$

where α and β are user-defined parameters to control the tradeoff between the two terms. Edge placement error (EPE) measures the manufacturing distortion by the difference of edge placement between the final image and the target image under nominal process condition. EPE may cause yield impact if its value is larger then a certain threshold th_{epe} and this is referred to as a violation. Process variability band (PV Band) [69] measures the layout sensitivity to process variations, which indicates a range of feature edge placement

among possible lithography process variations.

When the gradient descent algorithm is applied, the solution converges to the local optimum of the objective function closest to the initial condition. Starting from a good initial solution gives us a better chance to obtain a good result. An intuitive initial solution is the target mask. Instead of using the target mask directly, we apply simple rule-based OPC [56] by adding sub-resolution assist features (SRAF) in line 2.

Because the mask M contains only binary values, the ILT problem is an integer nonlinear problem and difficult to solve. It is common to relax the binary constraint to convert the ILT problem into an unconstrained optimization problem. We adopt the sigmoid transformation [93] as Eq. (3.8), which has been shown to provide effective solution searching for gradient descent:

$$M = sig(P) = \frac{1}{1+e^{-\theta_M \cdot P}} \quad , \theta_M \text{: steepness.} \tag{3.8}$$

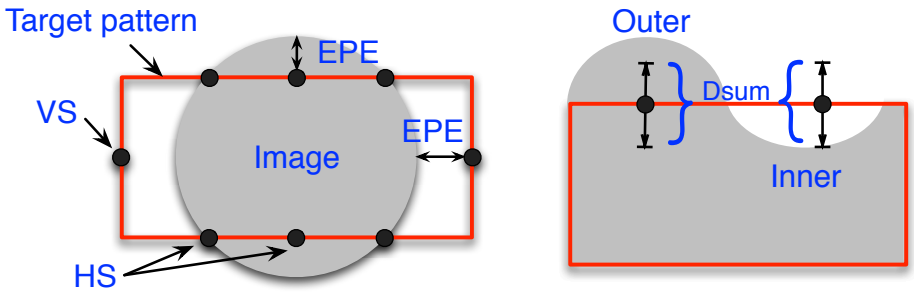
The relaxed variable P is therefore unbounded. Line 3 and line 7 in Alg. 6 perform the variable transformation based on the above definition.

In our gradient descent, we start from an initial mask solution and iteratively approach the optimum solution in the direction of the negative gradient of F with the number proportional to $stepSize$ (line 6). In order to directly calculate the gradient, F must be a differentiable function. We will discuss in Section 3.2.2∼3.2.4 how to define F properly and derive the closed form of its gradient. The optimization is repeated until an user-defined iteration threshold th_{iter} is reached or the solution converges to a local optimum. The

local optimum can be determined when the gradient becomes zero. Since each pixel inside the mask has its own gradient, we calculate the root mean square (RMS) of gradients of all pixels and exit the loop when it is less then a tolerance value th_g. We further improve the solution quality by exploring multiple local minima. Our implementation integrates the jump technique [93], where the step size will be adjusted to encourage searching the solution from different local minima.

3.2.2 Design Target Formulation based on EPE

In this section, we focus on the first half part of Eq. (3.7) for design target optimization. Although EPE is a common criterion to evaluate image contour, none of existed ILT approaches optimize EPE directly. Here, we propose an exact objective formulation for EPE minimization. Fig. 3.3 (a) gives an example of how EPE is measured. Measured points are sampled along the boundary of the target patterns, which includes a set of samples on horizontal edges (HS) and a set of samples on vertical edges (VS). We observe that the image distortion is continuous, producing either inner image edges or outer image edges as shown in Fig. 3.3 (b). Therefore, we can sum up the image difference as $Dsum$ within the range of the EPE constraint th_{epe}. The mathematical form is defined by Eq. (3.9).



(a) (b)

Figure 3.3: EPE measurement illustration.

$$Dsum_{i,j} = \sum_{k=j-th_{epe}}^{j+th_{epe}} D_{ik} \quad \text{, if } (i,j) \in HS$$
$$Dsum_{i,j} = \sum_{k=i-th_{epe}}^{i+th_{epe}} D_{kj} \quad \text{, if } (i,j) \in VS \tag{3.9}$$

where

$$D = (Z_{nom} - Z_t)^2 \tag{3.10}$$

We can then determine if there is an EPE violation based on Eq. (3.11). Again, since we need to formulate a differentiable equation, this threshold model is approximated by the sigmoid function with a steepness of θ_{epe}.

$$\text{EPE Violation} = \begin{cases} 0 & \text{if } Dsum < th_{epe} \\ 1 & \text{if } Dsum \geqslant th_{epe} \end{cases} \tag{3.11}$$

By checking $Dsum$ at all sample points $\{HS, VS\}$, we obtain the objective function for EPE minimization and its gradient as follows:

$$F_{epe} = \sum_{(i,j)\in HS} sig(Dsum_{i,j}) + \sum_{(i,j)\in VS} sig(Dsum_{i,j}) \tag{3.12}$$

$$\begin{aligned}\nabla F_{epe} &= \frac{\partial F_{epe}}{\partial p(x,y)} \\ &= \sum_{(i,j)\in HS} \frac{\partial sig(Dsum_{i,j})}{\partial p(x,y)} + \sum_{(i,j)\in VS} \frac{\partial sig(Dsum_{i,j})}{\partial p(x,y)}\end{aligned} \tag{3.13}$$

The closed form of the former part of Eq. (3.13) can be derived as Eq. (3.14), similarly for the later part.

$$\begin{aligned}&\sum_{(i,j)\in HS} \frac{\partial sig(Dsum_{i,j})}{\partial p(x,y)} \\ &= \sum_{(i,j)\in HS} \theta_{epe} \cdot sig(Dsum_{i,j})(1 - sig(Dsum_{i,j})) \sum_{k=j-th_{epe}}^{j+th_{epe}} \frac{\partial D_{ik}}{\partial p(x,y)}\end{aligned} \tag{3.14}$$

where

$$\begin{aligned}\frac{\partial D_{ik}}{\partial p(x,y)} &= \frac{\partial (Z_{nom}(i,k) - Z_t(i,k))^2}{\partial p(x,y)} \\ &= 2\theta_Z\theta_M \times (Z_{nom}(i,k) - Z_t(i,k)) Z_{nom}(i,k)(1 - Z_{nom}(i,k)) \\ &\quad \times \{[M(i,k) \otimes H^*_{nom}(i,k)] H_{nom}(i-x, k-y) \\ &\qquad + [M(i,k) \otimes H_{nom}(i,k)] H^*_{nom}(i-x, k-y)\} \\ &\quad \times M(i,k)(1 - M(i,k)).\end{aligned} \tag{3.15}$$

Here H^*_{nom} denotes the conjugate transpose of the kernel matrix H_{nom}. The derivation of Eq. (3.15) can be found in Appendix 1.

Note that the complexity of the gradient calculation is proportional to the size of the sample points $|HS|+|VS|$. If the target patterns are very complicated, the sample points would increase, and so does the computational time.

3.2.3 Design Target Formulation based on Image Difference

To improve the complexity of gradient calculation, we propose another objective formulation for design target optimization. The concept is to minimize the image difference (id) between the nominal image and the target image, as shown in Eq. (3.16).

$$F_{id} = \sum_{i=1}^{N}\sum_{j=1}^{N}(Z_{nom}(i,j) - Z_t(i,j))^{\gamma} \tag{3.16}$$

where γ is used to control the weight of the impact made by the image difference. According to Appendix 1, the gradient can be derived as:

$$\begin{aligned}\nabla F_{id} = \gamma\theta_Z\theta_M \cdot \{&H_{nom} \otimes [(Z_{nom} - Z_t)^{\gamma-1} \odot Z_{nom} \odot (1 - Z_{nom}) \odot (M \otimes H^*_{nom})] \\ &+ H^*_{nom} \otimes [(Z_{nom} - Z_t)^{\gamma-1} \odot Z_{nom} \odot (1 - Z_{nom}) \odot (M \otimes H_{nom})]\} \\ &\odot M \odot (1 - M)\end{aligned} \tag{3.17}$$

The quadratic form ($\gamma = 2$) of Eq. (3.16) has been used in previous ILT studies. We find that when performing the co-optimization of design target and process window, setting different γ can help make a trade-off between these two objectives. In our implementation, γ is set as 4.

3.2.4 Co-optimization for Design Target and Process Window

PV Band is the area between the outermost printed edge and the innermost printed edge among all process conditions. However, the outermost/innermost edge may be formed by more than one process condition

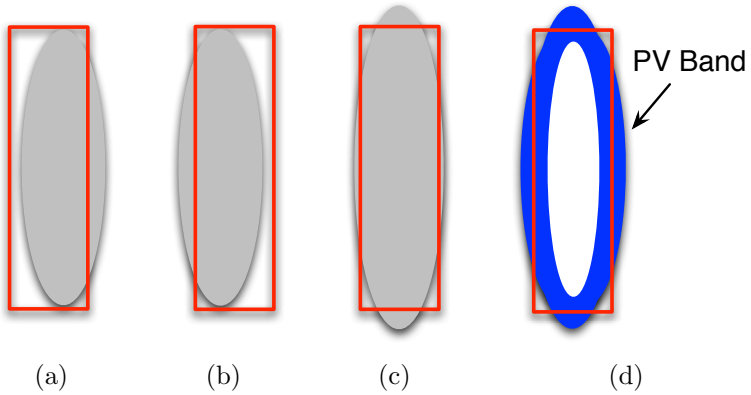


Figure 3.4: PV Band calculation. (a)∼(c) Printed images under different process conditions. (d) Resulted PV Band.

[69]. As illustrated in Fig. 3.4, the calculation of PV Band requires a series of boolean operations through all possible printed images. However, these boolean operations are difficult to model with a continuous form.

Therefore, we try to minimize the difference between possible images and the target image, as defined in Eq. (3.18) where N_p is the number of possible process conditions. With this formulation, we expect that inner edges and outer edges can be optimized toward the target edges which reduces the overall PV Band.

$$F_{pvb} = \sum_{k=1}^{N_p} (Z_k - Z_t)^2 \tag{3.18}$$

By combining Eq. (3.12) and Eq. (3.16) with Eq. (3.18), we can obtain the following objective functions that optimize design target and process window simultaneously. Both of the two functions are applied into Alg. 6 as

$MOSAIC_{exact}$ and $MOSAIC_{fast}$ respectively, and evaluated in Section 3.3.

$$F_{exact} = \alpha F_{epe} + \beta F_{pvb} \tag{3.19}$$

$$F_{fast} = \alpha F_{id} + \beta F_{pvb} \tag{3.20}$$

3.2.5 Speedup for Kernel Convolution

The gradient calculation requires a large amount of computational efforts from convolution operations, which is the main overhead of our approaches. We transform the non-quadratic form of Eq. (3.2) into Eq. (3.21) based on the properties of convolution, associativity with scalar multiplication and distributivity. With the new formulation of the kernel function, we can precompute H by combing all kernel models without losing the accuracy. This reduces the convolution operations by N_h times and significantly improves the efficiency of our approaches.

$$\begin{aligned} M \otimes H &= \sum_{k=1}^{N_h} w_k \cdot (M \otimes h_k) = \sum_{k=1}^{N_h} M \otimes (w_k \cdot h_k) \\ &= M \otimes \sum_{k=1}^{N_h} w_k \cdot h_k \end{aligned} \tag{3.21}$$

3.3 Experimental Results

Our ILT methods are implemented in C/C++ and tested on Linux machine with 3.4 GHz CPUs and 32 GB memory. We adopt the optical parameters from [3], with 193nm wavelength, a defocus range of $\pm 25nm$ and a

dose range of ±2%. Ten benchmarks released by IBM for the ICCAD 2013 contest [3] are tested, which represent the most challenging shapes to print. Each benchmark is a layout clip of 32nm M1 layer, with a size of $1024nm \times 1024nm$. The resolution of the pixelated mask is 1nm per pixel. EPE constraint th_{epe} is set as 15nm. EPE sample points are measured every 40nm along the pattern boundaries.

The parameters α and β in our objective functions are set based on the scoring function provided in [3] as follows:

$$\begin{aligned} \text{Minimize: } Score = Runtime + 4 \times PV\ \ Band + 5000 \times \#EPE \\ + 10000 \times ShapeViolation \end{aligned} \quad (3.22)$$

where $ShapeViolation$ is based on the existence of holes in the final contour. All our results produce zero $ShapeViolation$.

We compare our results with the top 3 winners of the ICCAD 2013 contest, where those approaches are also designed to optimize Eq. (3.22). The results are shown in Table 3.2 in terms of the number of EPE violations (#EPE), the area of process variability band (PVB), and $Score$. With the given scoring function, our approaches successfully achieve the best result (lowest score). Table 3.3 shows the runtime comparison of different OPC approaches. Note that the compared approaches are run on a different machine (2.65GHz CPU) from ours. However, we can still see that the runtime of MOSAIC˙fast is around the same scale as the contest results. Moreover, runtime only accounts for a small portion of the overall score, which accounts 0.12%

for MOSAIC˙fast and 0.75% for MOSAIC˙exact, respectively. Examples of our OPC result can be seen in Fig. 3.5.

Table 3.2: Comparison with the winners of the ICCAD 2013 contest.

Testcases		1st place			2nd place			3rd place			MOSAIC_fast			MOSAIC_exact		
Name	Pattern Area	#EPE	PVB	Score	#EPE	PVB	Score	#EPE	PVB	Score	#EPE	PVB	Score	#EPE	PVB	Score
B1	215344	0	65743	263578	6	57190	259242	2	70014	290329	6	58232	263246	9	56890	274267
B2	169280	1	53335	218659	13	45776	248589	0	58927	235838	10	47139	238812	4	48312	214493
B3	213504	25	143993	701266	39	90493	557459	35	106676	602009	59	82195	624101	52	84608	600955
B4	82560	0	31654	127030	14	24276	167591	1	38401	158891	1	28244	118298	3	24723	115161
B5	281958	0	65529	262378	16	55754	303505	4	69796	299394	6	56253	255327	2	56299	237363
B6	286234	1	62164	254086	18	49059	286718	0	59315	237351	1	50981	209238	1	49285	204224
B7	229149	0	51098	204787	8	43663	215134	8	56972	268241	0	46309	185475	0	46280	186761
B8	128544	0	25802	103447	0	23810	95771	0	26106	104504	2	22482	100186	2	22342	100031
B9	317581	2	74931	310008	15	62164	324225	12	78781	375533	6	65331	291646	3	62529	268138
B10	102400	0	18433	73904	0	19585	78829	0	18579	74376	0	18868	75703	0	18141	73276
Ratio				1.11			1.12			1.16			1.04			1.00

Pattern Area/PVB unit: nm^2

Table 3.3: Runtime comparison with the winners of the ICCAD 2013 contest.

Testcases	1st place	2nd place	3rd place	MOSAIC_fast	MOSAIC_exact
B1	606	482	273	318	1707
B2	319	485	130	256	1245
B3	294	487	305	321	2523
B4	414	487	287	322	1269
B5	262	489	210	315	2167
B6	430	482	91	314	2084
B7	395	482	353	239	1641
B8	239	531	80	258	663
B9	284	569	409	322	3022
B10	172	489	60	231	712
Average	341.5	498.3	219.8	289.6	1703.3

unit: *second*

3.3.1 Convergence of Gradient Descent

We further investigate the convergence of our gradient descent based ILT. In our experiments of Alg. 6, the maximum iteration number th_{iter} is 20 and the optimization is stopped at $th_g = 0.015$. Fig. 3.6 shows the convergence curves of testcase B4 and B6. We can see that the number of EPE violations gradually decreases while PV Band goes the opposite. This is because EPE has higher weight in the objective function. In the first few iterations, the mask patterns are nearly non-printable, and thus the result is less stable. The patterns become printable after a few optimization procedures, which also reflects the increase of PV Band as more iterations applied. In general, the optimization can converge quite effectively within 20 iterations.

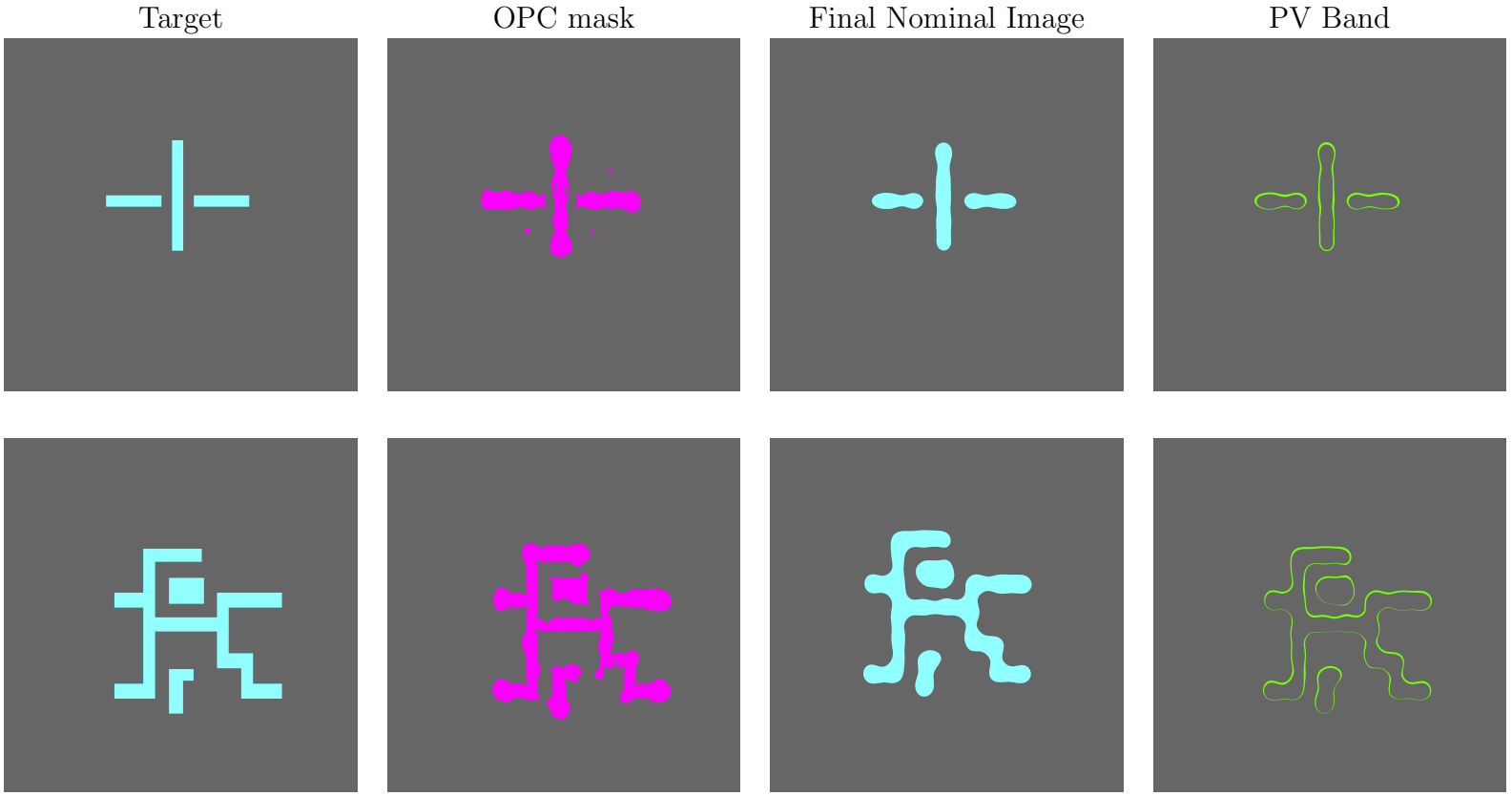


Figure 3.5: OPC result examples with MOSAIC˙exact. First row: B4; second row: B6.

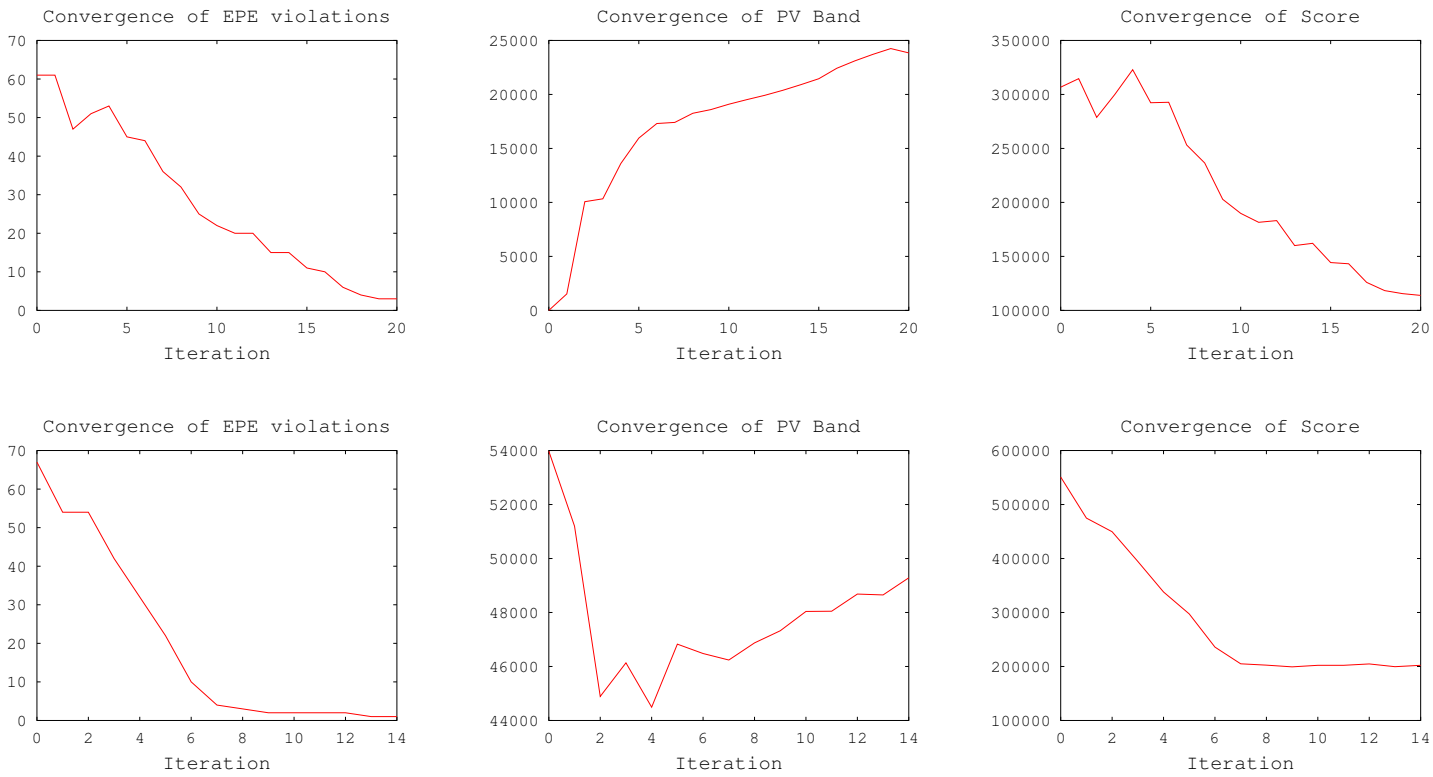


Figure 3.6: Convergence of the gradient decent with MOSAIC˙exact. First row: B4; second row: B6.

3.4 Summary

As the increasing challenges of semiconductor manufacturing, OPC becomes much more difficult. ILT based approaches have been a promising candidate for advanced technology nodes. We propose new mask optimizing solutions considering design target and process window simultaneously. Two approaches, MOSAIC˙exact based on exact EPE minimization and MOSAIC˙fast with efficient gradient computation are tested on 32nm designs. The results show that both of our approaches outperform the winners of the ICCAD 2013 contest.

Chapter 4

Accurate Lithography Hotspot Detection Based On PCA-SVM Classifier

With the continuous shrinking of technology nodes, layout patterns become more sensitive to lithography processes and degrade manufacturing yield. Lithography hotspots are forbidden topologies that need to be identified and eliminated during physical verification. Various design for manufacturing (DFM) techniques [64, 80] have been proposed to avoid these hotspots. In the meantime, there are resolution enhancement techniques (RET), such as optical proximity correction [84], phase-shift mask, and off-axis illumination, to improve the printability of problematic topologies. However, for deep sub-wavelength process, preventing lithography hotspots is still challenging and it requires accurate physical verification to identify these hotspots for yield improving.

In physical design and verification stages, the hotspot detection problem is to locate hotspots on a given layout with fast turn-around-time. Conventional lithography simulation [47, 67] obtains pattern images by complicated lithography models. Although it is accurate, full-chip lithography simulation is computational expensive, and thus cannot provide quick feedback to guide

the early physical design stages.

Recently, pattern matching based [44, 86] and machine learning based [21, 22, 24, 53, 72, 87] hotspot detection have become popular candidates. In pattern matching based approaches, a hotspot pattern is defined by its geometric characteristics, and the detection process involves matching the hotspot patterns with all layout patterns. This method relies on a set of pre-defined hotspot patterns, and patterns outside of the scope of this set may all be viewed as non-hotspots. Defining too many hotspot patterns would lead to over-estimation and over-optimization; while defining too few would limit the design space too aggressively. Although pattern matching based methods are accurate and fast, how to properly define hotspot patterns is still the main issue. In machine learning based approaches, a regression model is constructed according to a given training data, which includes hotspot and non-hotspot patterns. The model is then used to identify hotspots on a given testing layout. Machine learning based approaches enlarge the possible topologies for hotspots, therefore can improve the detection rate. However, it also increase the false alarms, which means some reported hotspots are not real hotspots.

Effective representation of layout data is essential for hotspot detection problem and there have been several encoding methods proposed. Kahng et al. presented an early hotspot detection [44] that builds a graph for the full layout to reflect pattern-related CD variation. This method depends on a limited set of CD variation evaluation methods, and thus false alarms may be generated. Yu et al. proposed a DRC-based hotspot detection [86] by

extracting critical topological features and modeling them as design rules. How to extract critical design rules is a crucial process for its performance because excessive rules would lead to numerous false alarms, while too few rules would lead to missed real hotspots. Range pattern [76, 79] is proposed to incorporate process-dependent specifications, which then can be used to identify hotspots by performing a string matching. Recently, Support Vector Machine (SVM) has become a popular data learning model for hotspot detection. Drmanac et al. [24] utilize Support Vector Machine (SVM) to train patterns represented by the histogram extracted from pixel-based layout images. In [72], layout density-based metrics are extracted to train the SVM kernel. A hybrid pattern matching and machine learning based approach [22] is proposed to take the advantages of both techniques.

In this chapter, we propose a high performance hotspot detection approach based on PCA-SVM classifier. Principle component analysis (PCA) is a technique for feature extraction and data reduction; combining PCA with SVM helps to improve the detection accuracy significantly. Besides, our approach integrates the advantages of pattern matching and data learning, where pattern matching techniques enable high accuracy and data learning algorithms provide high flexibility to adapt to new lithography processes and rules. The main contributions include:

- We propose a multi-level PCA-SVM based data learning flow that can extract critical layout information through mathematical analysis.

- We present a two-stage hierarchical data clustering approach to partition the layout data, such that irrelevant data can be processed by different classifiers for both efficiency and accuracy improvement.
- We apply several data compression techniques to enhance the performance of PCA-SVM, including data sampling for hotspot/non-hotspot imbalance, and dimension reduction for encoded layout data.
- The experimental results show that our approach effectively maximizes accuracy and minimizes false alarms at the same time, where more than 80% of hotspots on all given testing layouts can be identified successfully.

The rest of the chapter is organized as follows. We will first give the problem formulation in Section 4.1. Our proposed approaches including hotspot model calibration and full layout detection will be explained in Section 4.2 and Section 4.3, respectively. Finally, we will show our experimental results and performance analysis in Section 4.4, followed by the summary in Section 4.5. The preliminary results of this work were reported at [32].

4.1 Problem Formulation

The hotspot detection problem can be formulated as follows. Given two sets of verified layout clips, a set of hotspots and a set of non-hotspots, construct a system/model that can be used to identify unknown hotspots on a testing layout. The objective is to increase the number of true hotspots (Hit) and decrease the number of false hotspots ($Extra$).

A sample of the input layout clip is shown in Fig. 4.1. A *Frame* corresponds to the ambit or context area associated to its center. A *Core*, if indicated, corresponds to the central location where a hotspot appeared; otherwise the clip is free of hotspots. When given a testing layout with unknown hotspots, the hotspot detection engine should report all possible hotspot locations. However, excessive false hotspots reported would cause over-optimization for the later hotspot fixing stage, thus should be minimized.

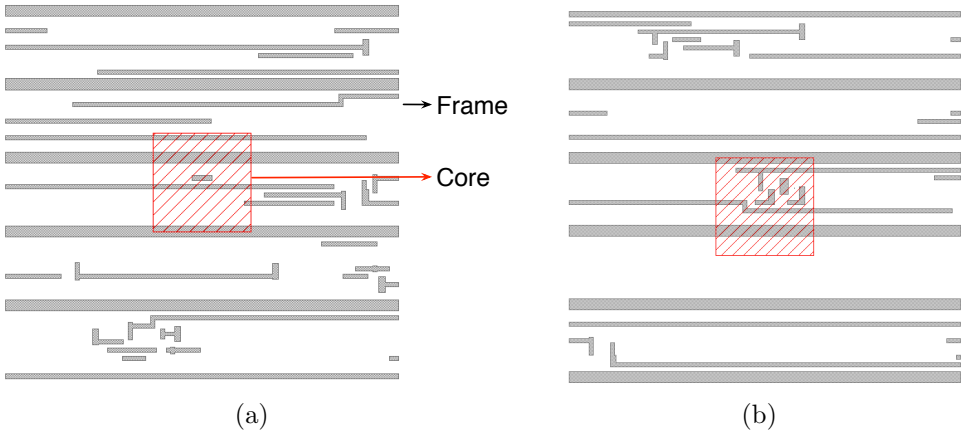


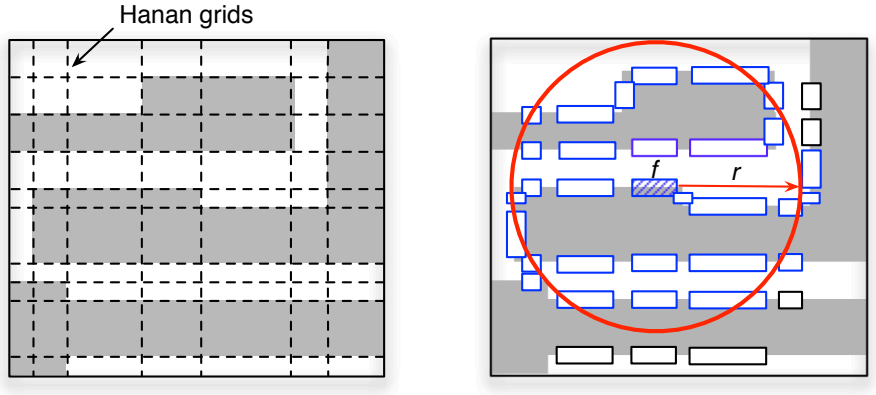
Figure 4.1: Examples of hotspot patterns marked in red. (a) Hotspot resulted from 1D patterns only. (b) Hotspot resulted from complex 1D and 2D patterns.

4.1.1 Layout Pattern Representation

A layout pattern becomes a hotspot not only because of the shape itself, but also because of the combined impact of its neighboring patterns. One fundamental step for the hotspot detection problem is to represent layout patterns with certain format that can well describe the layout environment. We adopt the concept of the fragmentation based context characterization [20] to

encode the layout patterns. This characterization method provides important layout information that is sufficient to describe a hotspot/non-hotspot, including pattern shapes, the distance between patterns, corner information (convex or concave), and etc.

Fgiure 4.2 (a) shows the contour of three layout patterns and their corresponding Hanan grids. Fragments are generated based on these grids as shown in (b). For each fragment f, an effective radius r is defined to cover the neighboring fragments which need to be considered in the context characterization of f. The radius r is process-dependent, which relates to how neighboring patterns can affect the fragment of interest f. We then extract all fragments f_r covered by r as shown in blue in Fig. 4.2 (b) and their properties. A complete representation of f includes the geometric characteristic of each



(a) (b)

Figure 4.2: Fragmentation based hotspot signature extraction. (a) Layout patterns and the Hanan grids shown in dashed lines. (b) Fragmentation based context characterization within the effective radius.

f_r, such as the length, corner, space, etc, which is stored as a vector for each fragment. In the following section, we refer to the characterized vector of a fragment as $Fragment\ Vector$ (FV).

The fragment generation in [20] is done by Calibre [4]. Since our hotspot detection flow is independent of Calibre, we generate fragments based on the Hanan grids in a $Frame$. All fragments inside a $Core$ are viewed as hotspot patterns, and the rest of the fragments are viewed as non-hotspot patterns.

4.2 Hotspot Model Calibration

The hotspot detection is essentially composed of two steps: hotspot model calibration with known patterns, and hotspot detection on testing layouts. In this section, we will introduce our approaches to calibrate accurate hotspot classification models, which will be used in the hotspot detection process.

4.2.1 Overall Data Calibration Flow

Fig. 4.3 shows our hotspot calibration flow. Given the training layout clips, we first decompose the layout patterns into small fragments based on Hanan grids, and collect a set of hotspot fragments and a set of non-hotspot fragments. We adopt the fragmentation based pattern characterization method [20] to encode fragments, in which each fragment is represented by a $Fragment\ Vector$ (FV). This characterization method provides layout information that is sufficient to describe a hotspot/non-hotspot environment,

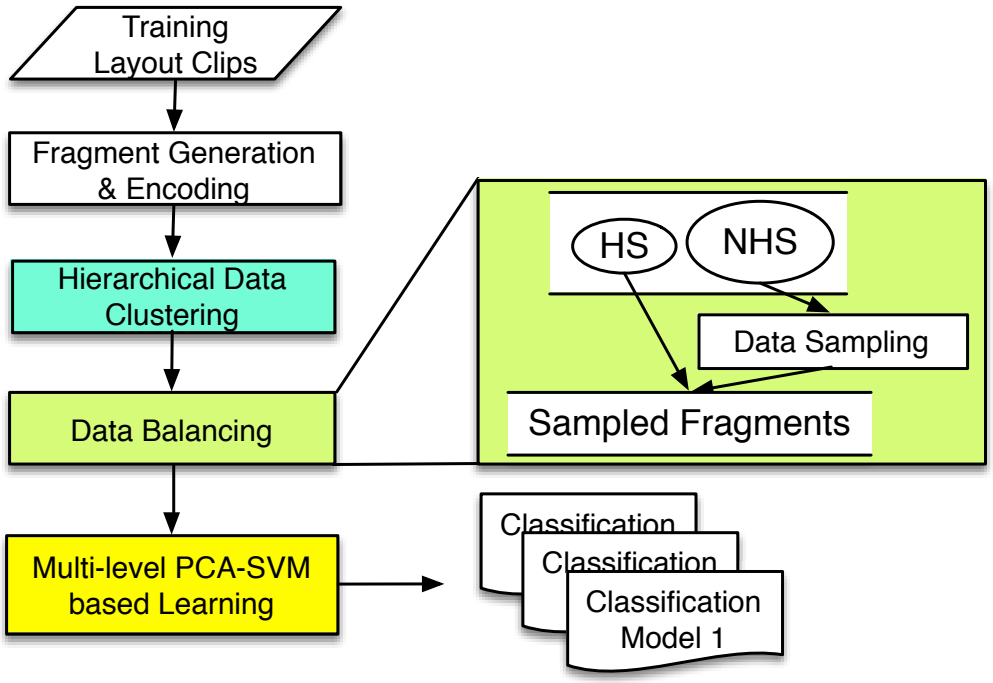


Figure 4.3: Hotspot model calibration flow.

such as pattern shapes, the distance between patterns, corner information (convex or concave), and etc. Next, we apply hierarchical data clustering to group similar fragments together based on their topological information (Section 4.2.2). Fragments in each cluster are sampled for data balancing (Section 4.2.3) and then sent to our PCA-SVM based learning process (Section 4.2.4). Finally, a set of hotspot classification models will be calculated for the use of the detection process.

4.2.2 Hierarchical Layout Data Clustering

The main objective of the hotspot calibration process is to build a model that can distinguish hotspots and non-hotspots. We observe that the accuracy of the calibrated model highly depends on the simplicity of the data. If the training data is very complicated, finding a general rule to classify them would be difficult and inaccurate. Fig. 4.1 shows two hotspot examples highlighted in red slashed rectangles; the hotspot in (a) simply results from 1D patterns, while the one in (b) involves several 2D patterns. Putting these two types of data in one classifier is already a challenge, not to mention there are much more types of hotspots. Training all data in a single classifier not only is time-consuming but also degrades the classification performance. Therefore, we propose a two-stage hierarchical layout data clustering approach to group the training data (fragments) according to their topological information.

The first-level clustering try to cluster fragments by capturing the global view of the pattern environment; and the second-level clustering further cluster the fragments within each global cluster based on FV to reflect the local view. By applying our clustering approach, the whole calibration process can be done in a divide-and-conquer manner. We will show that this clustering approach helps to improve the model accuracy and reduce the overall runtime in our experimental results.

4.2.2.1 Global Pattern Matching based Clustering

In the first stage, we apply a pattern matching based clustering technique to provide a quick clustering results in a global view. A set of represented pattern types are pre-defined in our pattern matching engine. These pattern types are obtained by observing the common pattern combinations in the testing layout clips. In general, 1D and 2D patterns are separated, and some special 1D/2D shapes are defined.

We define an *impact region* based on the lithography process. For a fragment f, if there is only 1D patterns in its impact region, f is clustered as a 1D pattern. Specifically, we define a 1D-type pattern that includes one long feature as shown in Fig. 4.4 (a), and another 1D-type pattern that includes parallel 1D features as shown in (b). On the other hand, if there is a 2D pattern inside the impact region of a fragment f, it is clustered as a 2D pattern. Fig. 4.4 (c) shows the pattern defined by an L-shaped feature and a long feature, while (d) shows an mountain-shaped pattern.

Pattern matching is performed for each fragment to determine which pattern type a fragment belongs to. If no specific pattern type is found, this fragment is assigned to a default cluster. The pattern matching based clustering only requires scanning the fragments within the impact region, and thus it can be effeciently implemented.

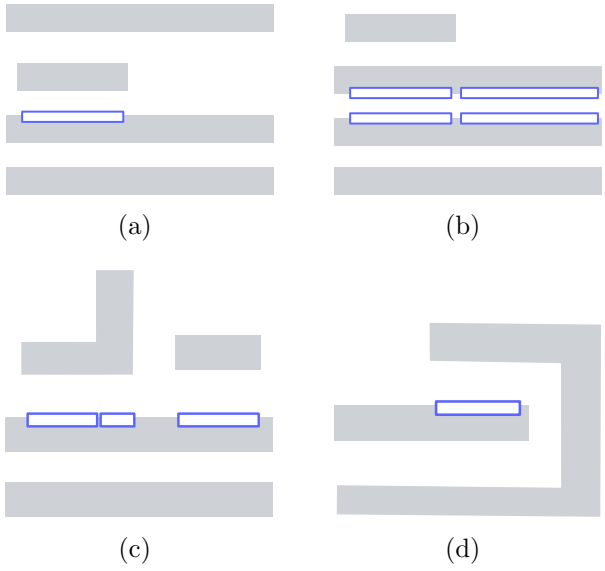


Figure 4.4: Pattern types. (a) 1D pattern including one long feature; (b) 1D pattern including two parallel long features; (c) 2D pattern including one long feature; (d) 2D mountain-shaped pattern.

4.2.2.2 Local K-means Clustering

Once the pattern matching based clustering is done, we further apply local clustering by k-means clustering [46] for each cluster obtained in the first stage. Given a set of N data points in d-dimensional space $\mathbf{R}^d$, k-means clustering partition the points into k disjoint subsets S. The objective is to minimize the sum of mean squared distance within each cluster:

$$\min : \sum_{i=1}^{k} \sum_{n \in S_i} \|x_n - \mu_i\|^2 \tag{4.1}$$

, where x_n is a vector representing the n-th data point, and μ is the mean of points in S_i. Fig. 4.5 shows an example of dividing points into five clusters in a 2-dimensional plane. By mapping FVs with d elements to d-dimensional

points, we can directly apply k-means clustering to partition the fragments inside each global cluster based on their difference of geometrical properties. Since Eq. (4.1) minimizes the sum of mean squared distance, each dimension of a data point should be at the same scale. In our implementation, we normalize each elements of FV before applying k-means clustering.

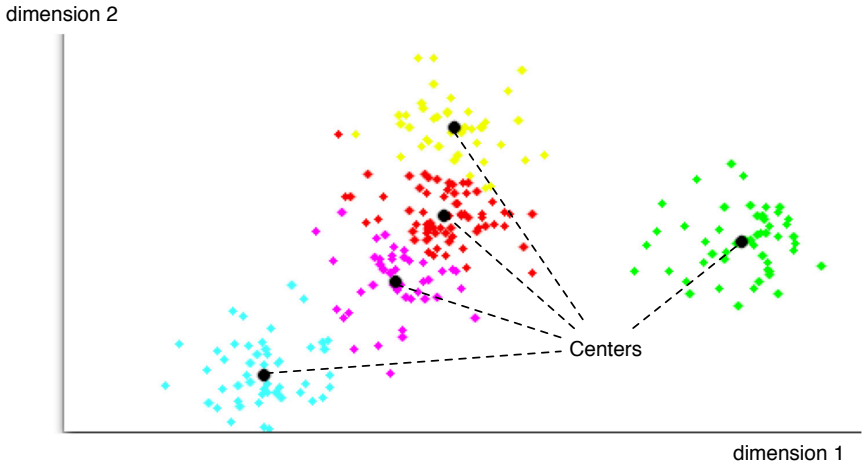


Figure 4.5: K-means clustering with $k = 5$.

4.2.3 Non-hotspot Data Balancing

There are numerous various-shaped patterns in a layout, and the problem is that non-hotspot patterns greatly outnumbers hotspot patterns [2]. For example, for a layout with hundreds of millions patterns per mm^2, the amount of hotspots may be less than 100. The imbalance between hotspot and non-hotspot data is called *imbalanced populations*, which critically affect the success of SVM learning [46]. In addition, since we decompose the layout patterns as $fragments$ and represent them by FVs, the size of the training data increases

rapidly. It is important to shrink the data size to speed up the later data learning process. To enhance both accuracy and efficiency, we propose a data sampling technique to reduce the number of non-hotpot data.

In simple random sampling, every element in the given data set has an equal chance to be chosen. However, this method is vulnerable to sampling error because the random selection may not reflect the real data distribution. In systematic sampling, on the other hand, elements of the given data set are first sorted in a certain order, and each element at a regular interval is selected. The difficulty of sampling FVs with systematic sampling is that the dimension d of each FV may be very high. In our experience, we may need a FV with $d = 250$ to well describe the property of a fragment. The sorting process for all fragments would be time-consuming. Besides, FVs are usually not evenly distributed in the d-dimensional space, which can result in over- or under- represented of the data.

Our data reduction approach utilize k-means clustering to group together data with similar geographical information. By doing this, we can choose the center of each cluster as the sampled data of the corresponding cluster, where the center is the mean of the data within a cluster. For example, in Fig. 4.5, the black circles are the data sampled for the five clusters. Setting a larger size of clusters can minimize the data difference within a cluster and reduce the sampling error; while setting a smaller size of clusters makes the training process faster with an average view of the data. By carefully choose the size of the cluster, we can keep the main characteristics of each cluster

without losing the sampling coverage.

4.2.4 Multi-level PCA-SVM based Classification

4.2.4.1 Dimension Reduction with PCA

PCA [43] is a statistical technique that analyzes a set of data composed of possibly inter-correlated variables. The goal is to extract the important information of the original data and represent the data as a new set of uncorrelated variables, called principle components. The number of principle components s is less than or equal to the number of the original variables. In Computer Vision field, the combination of PCA and SVM [18, 38] has been proven to improve the performance of pattern recognition. We apply PCA in front of our SVM process, which has the advantages of reducing the data size and increasing the hotspot classification accuracy.

The PCA problem is defined as follows. Given a data set $\mathbf{x} \in \mathbf{R}^d$, transform $\mathbf{x}$ into a new data set $\mathbf{y} \in \mathbf{R}^s$:

$$\begin{aligned} &y_{i,1} = A_{11}x_{i,1} + A_{12}x_{i,2} + ... + A_{1s}x_{i,s} \\ &y_{i,2} = A_{21}x_{i,1} + A_{22}x_{i,2} + ... + A_{2s}x_{i,s} \\ &... \\ &y_{i,s} = A_{s1}x_{i,1} + A_{s2}x_{i,2} + ... + A_{ss}x_{i,s} \\ \\ &\forall x_i = (x_{i,1}, x_{i,2}, ..., x_{i,d})^T \in \mathbf{x}, \qquad\qquad i = 1, ..., n \end{aligned} \tag{4.2}$$

such that each $y_i \in \mathbf{y}$ explains as much as possible of the variance in the original data set and that elements in $\mathbf{y}$ is uncorrelated. The correlation matrix A is is a $d \times d$ matrix, which defines the new coordinate system. Each i-th column

$A_i = (A_{i1}, A_{i2}, ...A_{is})$ is the i-th eigenvector of the data covariance matrix C.

$$C = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T \tag{4.3}$$

PCA starts from calculating the covariance matrix C and then solve the eigenvector problem:

$$CA_i = \lambda_i A_i, \quad i = 1, ..., n \tag{4.4}$$

to obtain eigenvalues λ and their corresponding eigenvectors. The eigenvector with the largest eigenvalue captures the most variation among the training vectors $\mathbf{x}$, while the eigenvector with the smallest eigenvalue has the least variation.

Geometrically, PCA enables us to calculate a projection of the data to a subspace formed by eigenvectors corresponding to the most dominant eigenvalues. By sorting the eigenvalues in descending order, we can choose the first s principle components to represent the original data. This allows us to reduce our high-dimensional FV into a much shorter and more unique vector.

4.2.4.2 SVM with Polynomial Kernel

SVM is a machine learning method for classification and learning tasks. In SVM, data vectors are mapped into a higher-dimensional space using a kernel function, and an optimal linear discrimination function in the space or an optimal hyperplane that fits the training data is built. The objective is to maximize the margin between the separating hyper-plane and the nearest data vectors from both classes.

We adopt C-type SVM [12, 17]. Given training vectors $x_i \in \mathbf{R}^d$, $i = 1, ..., n$, and an indicator vector $z \in \{1, -1\}$ for 2-class SVM, the problem formulation is briefed as follows.

$$\begin{array}{rl} \min\limits_{\alpha}: & \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \\ subject\ to & z^T\alpha = 0 \\ & 0 \leqslant \alpha_i \leqslant C, \qquad i = 1, ..., n \end{array} \tag{4.5}$$

where e is the vector of all ones, Q is an $n \times n$ positive semidefinite matrix, $Q_{ij} = z_i z_j K(x_i, x_j)$. The parameter C controls the trade-off between allowing training errors and forcing rigid separating margins. The kernel function K maps the data into the different space so a hyperplane can be used to do the separation. We use polynomial kernel function in our implementation, which achieves the best results in our experiments.

4.2.4.3 Multi-level Training for False Alarm Minimization

We obtain several clusters from the clustering step explained in Section 4.2.2 and train a kernel for each cluster individually. The fact that hotspot data is far less than non-hotspot data significantly affect the performance of SVM. We find that although our trained models can successfully identify true hotspots, numerous false alarms (*Extra*) are also reported. In order to reduce the number of false alarms, we adopt a multi-level self validation kernel structure. Conceptually, we verify our trained model using known data and collect false alarm information. These false alarms are fed into the training process in the next level, where the SVM model can focus on eliminating those false alarms.

Our multi-level kernel training flow is shown in Fig. 4.6. In Level 1, all data within a cluster is sent to the SVM kernel training process, where a classification model will be calculated. The classification model is tested with the same data used for training, and thus we can verify the performance of the model by the number of Hit and $Extra$. If the number of $Extra$ exceeds a certain threshold, we train another SVM kernel in the next level, where the input data will be only Hit and $Extra$ data. Eq. (4.6) is used to determine if the training process is continued, where α is a user-define parameter for the false alarm rate.

$$\frac{\#Extra}{\#Total\ Non\text{-}hotspots} > \alpha \tag{4.6}$$

The larger α is, the more Hit; however, the number of false alarms also goes up. In our implementation, α is set as 5%.

It is worth mentioning that the data in each cluster is independent. Therefore we can perform the kernel training and the later detection process in parallel. By taking advantage of multi-core machines, our approach can be more efficient, which is a practical feature for modern complex layouts.

4.3 Full Layout Hotspot Detection

Once the hotspot classification models are obtained in the training process, we can use these models to identify unknown hotspots on a given testing layout.

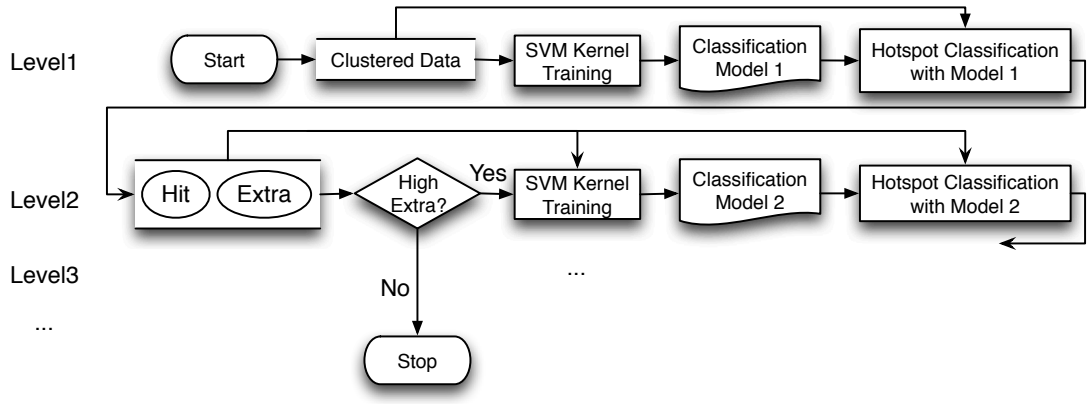


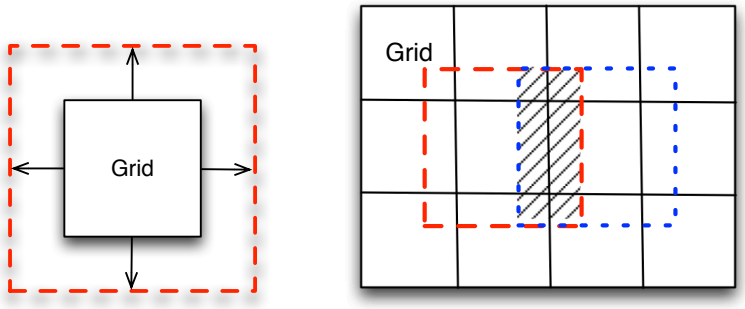
Figure 4.6: SVM based data learning flow.

4.3.1 Layout Scanning

Given a full layout, we need to generate fragments first and encode each fragment by FV according the geometric information in its nearby area. Constructing fragments and FV for the whole layout is time-consuming and impractical since hotspots are only formed in small regions. Therefore, we propose a layout scanning technique to perform our hotspot checking process in a more efficient way.

First, the layout is decomposed into grids. The grid size is process-dependent, which must be larger than the potential hotspot diameter to give sufficient information for FV. By default, we set the grid size the same as the frame size of the training clips. For each grid, we extract fragments and construct FV according to patterns inside the grid. Because the layout data outside of the grid is ignored at this time, we may miss some important information for fragments on the grid boundary. In order not to under-estimate

hotspots, we slightly enlarge the grid area whenever a grid is processed, as shown in Fig. 4.7 (a). In this way, we create an overlapped checking area between adjacent grids, which helps to increase hotspot identification accuracy. The red box and blue box in Fig. 4.7 (b) show two adjacent checking areas by enlarging their corresponding grids; the slashed area will be checked twice (one for the red box and one for the blue box) to ensure the result is not biased by the grid boundary.



(a) (b)

Figure 4.7: Layout scanning. (a) Enlarged checking area. (b) Slashes show the overlapped area between adjacent checking area.

4.3.2 Hotspot Identification Steps

In the hotspot detection process, we are required to identify hotspots on a given testing layout using the pre-built classification models. The hotspot identification flow is shown in Fig. 4.8. We first partition the layout into smaller grids and scan the layout on the grid base. The same fragment generation and data compression techniques as the training flow are applied. Each fragment to be tested will then be assigned into a specific cluster. Therefore,

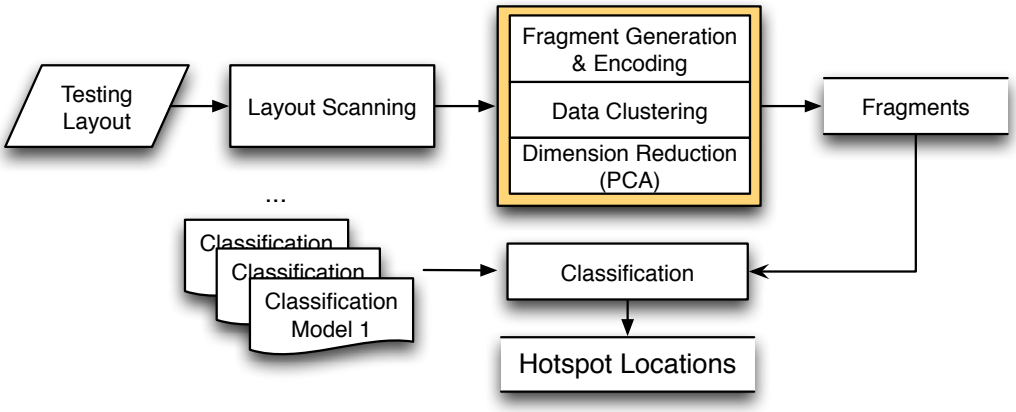


Figure 4.8: Full layout hotspot detection flow.

we can feed the fragments and FCs into their corresponding classification models obtained in the previous training process. According to our multi-level hotspot detection structure, a potential hotspot fragment must be identified by all classification models before they are reported. This helps to reduce the false alarms significantly.

4.4 Experimental Results

The proposed algorithms are implemented in C++ and tested on the machine with eight 3.0 GHz CPUs and 32 GB memory. The OpenMP [6] library is used for our parallel implementation. We apply the same setting of parameters in our approach for all benchmarks. The number of local clusters is set as 10; the maximum number of sampled non-hotspot centers within a cluster is 500; and the number of principle components for FV is 80.

We test our approach on the industrial benchmarks released in [2].

Table 4.1: ICCAD12 Benchmark statistics.

Tech	Training Layouts			Testing Layouts		
	Name	#HS	#NHS	Name	#HS	Area (mm^2)
32nm	MX_benchmark1	99	340	Array_benchmark1	226	12516
28nm	MX_benchmark2	174	5285	Array_benchmark2	498	106954
28nm	MX_benchmark3	909	4643	Array_benchmark3	1808	122565
28nm	MX_benchmark4	95	4452	Array_benchmark4	177	82010
28nm	MX_benchmark5	26	2716	Array_benchmark5	41	49583

Table 4.1 shows the statistics of five benchmarks, including 32nm and 28nm designs. The training layouts are the input of the hotspot calibration process, where hotspot and non-hotspot clips are given; while the testing layouts need to be verified by our hotspot detection flow to report the locations of identified hotspots. The number of total hotspot clips and non-hotspot clips are shown by $\#HS$ and $\#NHS$, respectively. According to the definition in [2], a reported hotspot is a Hit if it overlaps a real hotspot in the testing layout, otherwise it is an $Extra$. Here we define two important criteria to evaluate the performance of hotspot identification as shown in Eq. (4.7) and (4.8). Both terms should be maximized.

$$Accuracy = \frac{\#Hit}{\#HS} \tag{4.7}$$

$$H/E\ Ratio = \frac{\#Hit}{\#Extra} \tag{4.8}$$

Table 4.2 shows our results compared with [87]. Note that although [53]

Table 4.2: Result comparison with [87].

Testing Layout	Methods	Accuracy	H/E ratio	CPU (s)
Array_benchmark1	[12]	94.69%	0.143	$38.1s^*$
	Ours	80.97%	0.253	$63s^+$
Array_benchmark2	[12]	98.20%	0.041	$3m54s^*$
	Ours	81.12%	0.041	$34m57s^+$
Array_benchmark3	[12]	91.88%	0.123	$14m58s^*$
	Ours	90.93%	0.098	$29m42s^+$
Array_benchmark4	[12]	85.94%	0.045	$5m56s^*$
	Ours	87.01%	0.057	$13m8s^+$
Array_benchmark5	[12]	92.86%	0.032	$20s^*$
	Ours	80.49%	0.049	$8m26s^+$
Overall Impr.		-9.0%	27.17%	

* 2 Intel Xeon 2.3 GHz CPUs with 64 GB memory.
$^+$ 8 Intel Xeon 3.0 GHz CPUs with 32 GB memory.

also utilizes ICCAD12 benchmarks, their approach does not process a full layout but layout clips. Because of this fundamental difference, we cannot provide a fair comparison with [53]. From Table 4.2, we can see that our approach (*Ours*) steadily identifies more than 80% hotspots on all benchmarks, and maintain good H/E ratio at the same time. H/E ratio includes the information of both *Hit* and *Extra*. Since the hotspot detection problem requires both *Hit* maximization and *Extra* minimization, H/E ratio can more generally represent the overall performance. On average, our approach improves H/E ratio by 27.17% compared with [87].

The CPU time in Table 4.2 is the overall runtime including training and detection process. Table 4.3 shows the training time and detection time

Table 4.3: Runtime breakdown.

CPU time	Benchmarks				
	B1	B2	B3	B4	B5
Training	55s	29m4s	23m34s	11m14s	7m21s
Detection	8s	5m53s	6m8s	1m54s	1m5s

of our approach. It can be seen that the runtime spent on prediction is relatively low. In real application, the training process takes one-time effort to build the classification model. Then the obtained models can be repeatedly used for layouts with the same process parameters. It is worthwhile to obtain an accurate model with affordable runtime effort considering the model determines the detection performance and is built only once.

4.4.1 Performance Analysis of Non-hotspot Data Balancing

In Section 4.2.3, we introduce our data sampling technique for non-hotspots to alleviate the imbalance between hotspot and non-hotspot data. We adjust different sampling rates as Eq. (4.9) and see how the sampled data affects the results.

$$Sampling\ Rate = \frac{\#\text{Sampled fragments for the training process}}{\#\text{Total fragments}} \tag{4.9}$$

Fig. 4.9 shows the results of Array_benchmark5 with different sampling rates, where the x-axis is the sampling rate and the y-axis represents the values of the accuracy and the H/E ratio. One can observe that when the sampling rate gets higher, the results have the trend of lower accuracy and

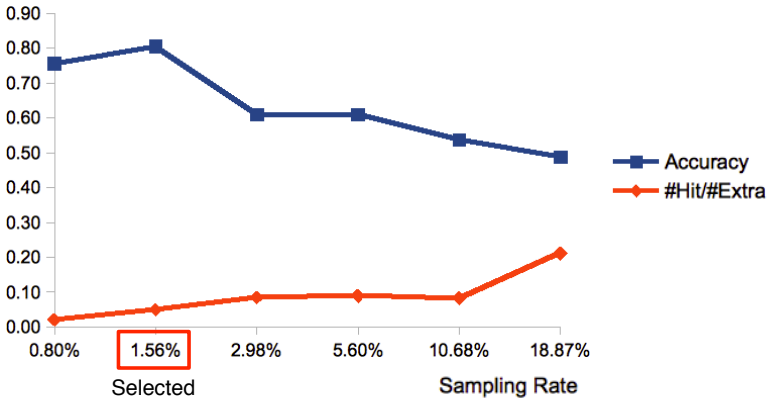


Figure 4.9: Results comparison with different sampling rates for Array benchmark5.

higher H/E ratio. This is because our training process needs to ensure the detection accuracy by the multi-level SVM kernels. When the number of data is large, our training process would generate stricter detection models to prevent false alarms. The trend of increasing H/E ratio and decreasing accuracy reflects the effect of the stricter models. The sampling rate needs to be decided properly to maintain a good trade-off between the accuracy and the H/E ratio. In our implementation, we set 80% accuracy as our main optimization objective, and then higher H/E ratio is considered. As a result, the 1.56% compression rate in Fig. 4.9 is selected as our final parameter.

4.4.2 Performance Analysis of PCA-based SVM

In order to understand the impact on SVM results by applying PCA, we implement two versions of our approach, one uses the presented PCA-SVM (w/ PCA), and the other uses typical SVM (w/o PCA). The maximum length

Table 4.4: Comparison of results with PCA and without PCA applied.

Benchmark	w/o PCA			w/ PCA		
	Accuracy	H/E ratio	CPU (s)	Accuracy	H/E ratio	CPU (s)
B1	80.09%	0.217	69	80.97%	0.253	63
B2	81.73%	0.041	2005	81.12%	0.041	2097
B3	85.90%	0.074	1934	85.40%	0.092	1782
B4	87.57%	0.023	814	87.01%	0.057	788
B5	82.93%	0.036	496	80.49%	0.049	506
Average	1	1	1	0.99	1.45	0.97

of FV without PCA is 250, while the maximum length of FV with PCA is 80.

Table 4.4 shows the comparison of the five benchmarks in terms of accuracy, H/E ratio, and CPU Time. We can see that the difference on the accuracy is little, showing that reducing the vector dimension does not lose critical information. On the other hand, the H/E ratio are significantly improved in most cases, showing that eliminating less-relevant information using PCA helps to reduce false alarms. The results show the effectiveness of PCA-SVM on reducing the false alarms and runtime, while maintaining the accuracy at the same time.

4.5 Summary

Lithography hotspots have a great impact on the manufacturing yield. Identifying the forbidden pattern topologies in the physical verification or early physical design stage has become a critical problem. We present a high performance hotspot detection approach based on PCA-SVM classifier. Several

techniques, including hierarchical data clustering, data balancing, and multi-level training, are provided to enhance performance of the proposed approach. Besides, our approach integrates the advantages of pattern matching and data learning, where pattern matching techniques enable high accuracy and data learning algorithms provide high flexibility to adapt to new lithography processes and rules. Our data clustering and data compression techniques help to improve the accuracy and reduce the false alarms. The experimental results show that our approach effectively maximizes accuracy and minimizes false alarms at the same time, where more than 80% of hotspots on all given testing layouts can be identified successfully.

Chapter 5

Conclusions and Future Works

In this dissertation, the lithography impact for design manufacturability is thoroughly studied, and the co-optimization of design performance and manufacturability in various physical design stages are presented. Our major contributions include:

- In Chapter 2, we tackle the challenge of enabling self-aligned double patterning and present the design methodologies to integrate SADP in placement, detailed routing, and post routing stages. To ensure the decomposability between standard cells, a SADP-aware legalization is applied to adjust cell placement with the minimum perturbation in Section 2.2. In Section 2.3, we perform simultaneous routing and mask assignment to guarantee the final route is free of coloring conflicts. In order to reduce the routability degradation by too much DFM aware rules, we propose a new design methodology to allow the router to locally fix weak patterns and maintain timing closure at the same time in Section 2.4. To further improve the design flexibility in advanced technology nodes, we study the complementary lithography with SADP and EBL in Section 2.5, and develop a min-cost max-matching based layout decomposition

to optimize mask and e-beam assignment.

- In Chapter 3, we present a new optical proximity correction algorithm based on inverse lithography technique considering both design target optimization and process window minimization. Two approaches, MOSAIC_exact based on exact EPE minimization and MOSAIC_fast with efficient gradient computation are developed.

- In Chapter 4, we discuss the difficulty of identify lithography hotspots during physical verification. A PCA-SVM based approach is proposed to efficiently and accurately predict hotspot patterns. The nature of the machine learning model allows our method to embrace fuzzy pattern matching for potential but unseen hotspots.

We have explained the challenges to deal with the lithography limit in advanced VLSI designs. We expect our work can motivate more follow-up research along the direction of co-optimization for design performance and manufacturability. To conclude the dissertation, we would like to point out some future research directions and issues:

While SADP has been widely used for modern designs, more advanced manufacturing processes will be required when semiconductor industries keep pushing pitch scaling. Researchers have begun to study the process issues for self-aligned quadruple patterning (SAQP) and general self-aligned multiple patterning (SAMP) that utilize multiple spacer depositions and trim masks to increase pattern resolution. Physical design methods that achieve

SAQP/SAMP enabled layouts would need to be studied. Other issues such as process variation, pin access, and pattern regularity will become more challenging and need to be addressed for SAQP/SAMP.

Next generation lithography techniques such as EUVL, Directed Self Assembly (DSA), and nanoimprint lithography can be further studied and evaluated as options for future manufacturing. Innovative CAD methodologies will need to be developed to handle their design challenges and make the mass production with these lithography techniques possible. Specifically, complementary lithography can be achieved by 193i lithography wit EBL, or EUVL, or DSA; each comes with pros and cons. For example, the optimization objective for EBL is beam shots minimization, while that for DSA is guiding templates satisfaction. How to determine a good option for complementary lithography and to make good trade-off between pattern quality and manufacturing cost are still open problems.

Appendix

Appendix 1

Formula Derivation in Chapter 3

The basic differential equations can be found as follows:

$$\begin{aligned}\frac{\partial sig(x)}{\partial x} &= \frac{\partial \frac{1}{1+e^{-\theta(x-t)}}}{\partial x} \\ &= \theta sig(x)[1 - sig(x)],\end{aligned} \tag{1.1}$$

and

$$\frac{\partial M(i,j) \otimes H(i,j)}{\partial p(x,y)} = H(i-x, j-y). \tag{1.2}$$

Let $p(i,j)$ be a pixel in the imaging system, where $1 \leqslant i,j \leqslant N$. We represent the variable definitions $M(i,j), H_{nom}(i,j), I_{nom}(i,j), Z_{nom}(i,j), Z_t(i,j)$ as m, h, i_A, z, z_t for simplification.

The partial differential of the light intensity i_A can be derived as follows:

$$\begin{aligned}\frac{\partial i_A}{\partial p(x,y)} &= \frac{\partial |m \otimes h|^2}{\partial p(x,y)} \\ &= \frac{\partial (m \otimes h)(m \otimes h^*)}{\partial p(x,y)} \times \frac{\partial m}{\partial p(x,y)} \\ &= [(m \otimes h^*)H(i-x, j-y) + (m \otimes h)H^*(i-x, j-y)] \\ &\quad \times \theta_M \cdot sig(p(x,y))(1 - sig(p(x,y))) \\ &= \theta_M \cdot m(x,y)(1 - m(x,y)) \\ &\quad [(m \otimes h^*)H(i-x, i-j) + (m \otimes h)H^*(i-x, j-y)].\end{aligned} \tag{1.3}$$

The derivation of Eq. (3.15) is as follows:

$$
\begin{aligned}
&\frac{\partial d}{\partial p(x,y)} \\
&= \frac{\partial (z - z_t)^2}{\partial p(x,y)} \\
&= 2(z - z_t)\frac{\partial z}{\partial p(x,y)} \\
&= 2(z - z_t)\frac{\partial sig(i_A)}{\partial p(x,y)} \\
&= 2(z - z_t)\theta_Z sig(i_A)(1 - sig(i_A))\frac{\partial i_A}{\partial p(x,y)} \\
&= 2\theta_Z\theta_M(z - z_t)z(1 - z) \cdot [(m \otimes h^*)H(i - x, i - j) + (m \otimes h)H^*(i - x, j - y)] \\
&\quad \cdot m(x,y)(1 - m(x,y)).
\end{aligned}
\tag{1.4}
$$

Given the above equation, the derivation of Eq. (3.17) is as follows:

$$
\begin{aligned}
\frac{\partial F_{id}}{\partial p(x,y)} &= \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial (z - z_t)^\gamma}{\partial p(x,y)} \\
&= \gamma\theta_Z\theta_M \sum_{i=1}^{N}\sum_{j=1}^{N}(z - z_t)^{\gamma-1}z(1 - z) \\
&\quad \times [(m \otimes h^*)H(i - x, i - j) + (m \otimes h)H^*(i - x, j - y)] \\
&\quad \times m(x,y)(1 - m(x,y)) \\
&= \gamma\theta_Z\theta_M \\
&\quad \cdot \{H_{nom} \otimes [(Z_{nom} - Z_t)^{\gamma-1} \odot Z_{nom} \odot (1 - Z_{nom}) \odot (M \otimes H^*_{nom})] \\
&\quad + H^*_{nom} \otimes [(Z_{nom} - Z_t)^{\gamma-1} \odot Z_{nom} \odot (1 - Z_{nom}) \odot (M \otimes H_{nom})]\} \\
&\quad \odot M \odot (1 - M).
\end{aligned}
\tag{1.5}
$$

Bibliography

[1] Cadence SOC Encounter [http://www.cadence.com/].

[2] ICCAD contest 2012 [http://cad_contest.cs.nctu.edu.tw/CAD-contest-at-ICCAD2012/problems/p3/p3.html].

[3] ICCAD contest 2013 [http://cad_contest.cs.nctu.edu.tw/CAD-contest-at-ICCAD2013/problem_c/].

[4] Mentor Graphics Calibre [http://www.mentor.com/].

[5] NanGate FreePDK45 Generic Open Cell Library [http://www.si2.org/openeda.si2.org/projects/nangatelib].

[6] OpenMP [http://www.openmp.org/].

[7] Accelerating physical verification with an in-design flow,. In *Synopsys White Paper*, 2009.

[8] George E. Bailey, Alexander Tritchkov, Jea-Woo Park, Le Hong, Vincent Wiaux, Eric Hendrickx, Staf Verhaegen, Peng Xie, and Janko Versluijs. Double pattern EDA solutions for 32nm HP and beyond. In *Proc. of SPIE*, volume 6521, 2007.

[9] Yongchan Ban, Kevin Lucas, and David Z. Pan. Flexible 2D layout decomposition framework for spacer-type double pattering lithography.

In *ACM/IEEE Design Automation Conference (DAC)*, pages 789–794, 2011.

[10] Shayak Banerjee, Kanak B. Agarwal, and Michael Orshansky. Methods for joint optimization of mask and design targets for improving lithographic process window. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 12(2):023014, June 2013.

[11] Yan Borodovsky. MPProcessing for MPProcessors. In *Maskless Lithography and Multibeam Mask Writer Workshop*, 2010.

[12] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proc. Computational Learning Theory*, pages 144–152, 1992.

[13] Szu-Yu Chen and Yao-Wen Chang. Native-conflict-aware wire perturbation for double patterning technology. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 556–561, November 2010.

[14] Minsik Cho, Yongchan Ban, and David Z. Pan. Double patterning technology friendly detailed routing. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 506–511, November 2008.

[15] Nicolas B. Cobb. Fast optical and process proximity correction algorithms for integrated circuit manufacturing. *PhD dissertation, University of California at Berkeley*, 1998.

[16] Nicolas B. Cobb and Yuri Granik. OPC methods to improve image slope and process window. In *Proc. SPIE*, pages 116–125, 2003.

[17] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Journal of Machine Learning*, 20(3):273–297, September 1995.

[18] O. Déniz, M. Castrillón, and M. Hernández. Face recognition using independent component analysis and support vector machines. *Pattern Recognition Letters*, 24(13):2153–2157, September 2003.

[19] Duo Ding, Jhih-Rong Gao, Kun Yuan, and David Z. Pan. AENEID: A generic lithography-friendly detailed router based on post ret data learning & hotspot prediction. In *ACM/IEEE Design Automation Conference (DAC)*, pages 795–800, 2011.

[20] Duo Ding, Andres J. Torres, Fedor G. Pikus, and David Z. Pan. High performance lithographic hotspot detection using hierarchically refined machine learning. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 775–780, 2011.

[21] Duo Ding, Xiang Wu, Joydeep Ghosh, and David Z. Pan. Machine learning based lithographic hotspot detection with critical-feature extraction and classification. In *IEEE International Conference on IC Design and Technology (ICICDT)*, pages 219–222, 2009.

[22] Duo Ding, Bei Yu, Joydeep Ghosh, and David Z. Pan. EPIC: Efficient prediction of IC manufacturing hotspots with a unified meta-classification

formulation. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 263–270, 2012.

[23] Martin Drapeau, Vincent Wiaux, Eric Hendrickx, Staf Verhaegen, and Takahiro Machida. Double patterning design split implementation and validation for the 32nm node. In *Proc. of SPIE*, volume 6521, 2007.

[24] Dragoljub Gagi Drmanac, Frank Liu, and Li-C. Wang. Predicting variability in nanoscale lithography processes. In *ACM/IEEE Design Automation Conference (DAC)*, pages 545–550, 2009.

[25] Yuelin Du, Qiang Ma, Hua Song, James Shiely, and Gerard Luk-Pat. Spacer-is-dielectric-compliant detailed routing for self-aligned double patterning lithography. In *ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2013.

[26] Yuelin Du, Hongbo Zhang, M.D.F. Wong, and Kai-Yuan Chao. Hybrid lithography optimization with E-Beam and immersion processes for 16nm 1D gridded design. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 707–712, 2012.

[27] Shao-Yun Fang, Szu-Yu Chen, and Yao-Wen Chang. Native-conflict and stitch-aware wire perturbation for double patterning technology. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 31(5):703–716, May 2012.

[28] Jo Finders, Micrea Dusa, and Stephen Hsu. Double patterning lithography: The bridge between low k1 ArF and EUV. *Microlithography World*, February 2008.

[29] Jhih-Rong Gao, Harshdeep Jawandha, Prasad Atkarc, Atul Walimbe, Bikram Baidya, and David Z. Pan. Self-aligned double patterning compliant routing with in-design physical verification flow. In *Proc. of SPIE*, volume 8684, 2013.

[30] Jhih-Rong Gao and David Z. Pan. Flexible self-aligned double patterning aware detailed routing with prescribed layout planning. In *ACM International Symposium on Physical Design (ISPD)*, pages 25–32, 2012.

[31] Jhih-Rong Gao, Xiaoqing Xu, Bei Yu, and David Z. Pan. MOSAIC: Mask optimizing solution with process window aware inverse correction. In *ACM/IEEE Design Automation Conference (DAC)*, 2014.

[32] Jhih-Rong Gao, Bei Yu, Duo Ding, and David Z. Pan. Accurate lithography hotspot detection based on PCA-SVM classifier with hierarchical data clustering. In *Proc. of SPIE*, volume 9053, 2014.

[33] Jhih-Rong Gao, Bei Yu, Ru Huang, and Pan David Z. Pan. Self-aligned double patterning friendly configuration for standard cell library considering placement. In *Proc. of SPIE*, volume 8684, 2013.

[34] Jhih-Rong Gao, Bei Yu, and David Z. Pan. Self-aligned double patterning layout decomposition with complementary E-Beam lithography.

In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 143–148, 2014.

[35] Rani S. Ghaida, Kanak B. Agarwal, Sani R. Nassif, Xin Yuan, Lars W. Liebmann, and Puneet Gupta. Layout decomposition and legalization for double-patterning technology. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 32(2):202–215, February 2013.

[36] Yuri Granik. Illuminator optimization methods in microlithography. In *Proc. of SPIE*, volume 5754, pages 217–229, 2005.

[37] Yuri Granik. Fast pixel-based mask optimization for inverse lithography. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 5(4):043002, December 2006.

[38] Ergun Gumus, Niyazi Kilic, Ahmet Sertbas, and Osman N. Ucan. Evaluation of face recognition techniques using PCA, wavelets and SVM. *Journal of Expert Systems with Applications*, 37(9):6404–6408, September 2010.

[39] Antonin Guttman. R-Trees: a dynamic index structure for spatial searching. In *Proc. SIGMOD Int. Conf. on Management of data*, pages 47–57, 1984.

[40] H. H. Hopkins. The concept of partial coherence in optics. pages A217: 408–432. Proceedings of the Royal Society of London, 1953.

[41] Chin-Hsiung Hsu, Yao-Wen Chang, and Sani Richard Nassif. Simultaneous layout migration and decomposition for double patterning technology. 30(2):284–294, February 2011.

[42] Ningning Jia and Edmund Y. Lam. Machine learning for inverse lithography: using stochastic gradient descent for robust photomask synthesis. *Journal of Optics*, 12(4):045601, April 2010.

[43] Ian T. Jollife. Principal component analysis. In *Springer-Verlag*, 1986.

[44] Andrew B. Kahng, Chul-Hong Park, and Xu Xu. Fast dual graph based hotspot detection. In *Proc. of SPIE*, volume 6925, 2006.

[45] Andrew B. Kahng, Chul-Hong Park, Xu Xu, and Hailong Yao. Layout decomposition for double patterning lithography. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 465–472, November 2008.

[46] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. 24(7):881–892, July.

[47] Juhwan Kim and Minghui Fan. Hotspot detection on Post-OPC layout using full chip simulation based verification tool: A case study with aerial image simulation. In *Proc. of SPIE*, volume 5256, 2003.

[48] Chikaaki Kodama, Hirotaka Ichikawa, Koichi Nakayama, Toshiya Kotani, Shigeki Nojima, Shoji Mimotogi, Shinji Miyamoto, and Atsushi Taka-

hashi. Self-aligned double and quadruple patterning-aware grid routing with hotspots control. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 267–272, 2013.

[49] Azalia Krasnoperova, James A. Culp, Ioana Graur, Scott Mansfield, Mohamed Al-Imam, and Hesham Maaty. Process window OPC for reduced process variability and enhanced yield. In *Proc. of SPIE*, volume 6154, 2006.

[50] David Lam, Dave Liu, and Ted Prescop. E-beam direct write (EBDW) as complementary lithography. In *Proc. of SPIE*, volume 7823, 2010.

[51] David Lam, Enden D. Liu, Michael C. Smayling, and Ted Prescop. E-beam to complement optical lithography for 1D layouts. In *Proc. of SPIE*, volume 7970, 2011.

[52] Lars Liebmann, David Pietromonaco, and Matthew Graf. Decomposition-aware standard cell design flows to enable double-patterning technology. In *Proc. of SPIE*, volume 7974, 2011.

[53] Sheng-Yuan Lin, Jing-Yi Chen, Jin-Cheng Li, Wan yu Wen, and Shih-Chieh Chang. A novel fuzzy matching model for lithography hotspot detection. In *ACM/IEEE Design Automation Conference (DAC)*, pages 68:1–68:6, 2013.

[54] Kevin Lucas, Christopher Cork, Alexander Miloslavsky, Gerard Luk-Pat, Levi Barnes, John Hapli, John Lewellen, Greg Rollins, Vincent Wiaux,

and Staf Verhaegen. Double-patterning interactions with wafer processing, optical proximity correction, and physical design flows. *Journal of Microlithography, Microfabrication and Microsystems*, 8(3), July 2009.

[55] Xu Ma and Gonzalo R. Arce. Generalized inverse lithography methods for phase-shifting mask design. 15(23):15066–15079, November 2007.

[56] Xu Ma and Gonzalo R. Arce. *Computational Lithography*. Wiley Series in Pure and Applied Optics, first edition, 2010.

[57] Yuansheng Ma, Jason Sweis, Chris Bencher, Huixiong Dai, Yongmei Chen, Jason P. Cain, Yunfei Deng, Jongwook Kye, and Harry J. Levinson. Decomposition strategies for self-aligned double patterning. In *Proc. of SPIE*, volume 7641, 2010.

[58] Yuansheng Ma, Jason Sweis, Hidekazu Yoshida, Yan Wang, Jongwook Kye, and Harry J. Levinson. Self-aligned double patterning (SADP) compliant design flow. In *Proc. of SPIE*, volume 8327, 2012.

[59] Pravin Madhani. Advanced manufacturing closure with calibre inroute. In *Globalpress Electronics Summit*, 2010.

[60] Rainer Mann, Ulrich Hensel, Vito Dai, and Jens Peters. Using synopsys ic compiler for dfm optimization at 28nm. In *SNUG*, 2011.

[61] Minoo Mirsaeedi, J. Andres Torres, and Mohab Anis. Self-aligned double-patterning (SADP) friendly detailed routing. In *Proc. of SPIE*, volume 7974, 2011.

[62] Kenichi Oyama, Eiichi Nishimura, Masato Kushibiki, Kazuhide Hasebe, Shigeru Nakajima, Hiroki Murakami, Arisa Hara, Shohei Yamauchi, Sakurako Natori, Kazuo Yabe, Tomohito Yamaji, Ryota Nakatsuji, and Hidetami Yaegashi. The important challenge to extend spacer DP process towards 22nm and beyond. In *Proc. of SPIE*, volume 7639, March 2010.

[63] L. Pain, M. Jurdit, J. Todeschini, S. Manakli, B. Icard, B. Minghetti, G. Bervin, A. Beverina, F. Leverd, M. Broekaart, P. Gouraud, V. De Jonghe, Ph. Brun, S. Denorme, F. Boeuf, V. Wang, and D. Henry. Electron beam direct write lithography flexibility for asic manufacturing an opportunity for cost reduction. *Proc. of SPIE*, 5751:35–45, 2005.

[64] David Z. Pan, Bei Yu, and Jhih-Rong Gao. Design for manufacturing with emerging nanolithography. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2(10):1453–1472, October 2013.

[65] Linyong Pang, Yong Liu, and Dan Abrams. Inverse lithography technology (ILT): What is the impact to the photomask industry? In *Proc. of SPIE*, volume 6283, 2006.

[66] Amyn Poonawala and Peyman Milanfar. Mask design for optical microlithography–an inverse imaging problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 16(3):774–788, March 2007.

[67] Ed Roseboom, Mark Rossman, Fang-Cheng Chang, and Philippe Hurat. Automated full-chip hotspot detection and removal flow for interconnect layers of cell-based designs. In *Proc. of SPIE*, volume 6521, 2007.

[68] Yijiang Shen, Ngai Wong, and Edmund Y. Lam. Level-set-based inverse lithography for photomask synthesis. *Optics Express*, 17(26):23690–23701, 2009.

[69] Juan A. Torres and C. N. Berglund. Integrated circuit DFM framework for deep subwavelength processes. In *Proc. of SPIE*, volume 5756, pages 39–51, 2005.

[70] Amr G. Wassala, Heba Sharafb, and Sherif Hammouda. Placement-aware decomposition of a digital standard cells library for double patterning lithography. In *Proc. of SPIE*, volume 8522, 2012.

[71] Alfred Kwok-Kit Wong. Resolution enhancement techniques. *SPIE Press*, 2001.

[72] Jen-Yi Wuu, Fedor G. Pikus, Andres Torres, and Malgorzata Marek-Sadowska. Rapid layout pattern classification. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 781–786, 2011.

[73] Guangming Xiao, Dong Hwan Son, Tom Cecil, Dave Irby, David Kim, Ki-Ho Baik, Byung-Gook Kim, SungGon Jung, Sung Soo Suh, and HanKu

Cho. E-beam writing time improvement for inverse lithography technology mask for full-chip. In *Proc. of SPIE*, volume 7748, 2010.

[74] Zigang Xiao, Hongbo Zhang, Yuelin Du, and Martin D. F. Wong. A polynomial time exact algorithm for self-aligned double patterning layout decomposition. In *ACM International Symposium on Physical Design (ISPD)*, pages 17–24, 2012.

[75] Zigang Xiao, Hongbo Zhang, Yuelin Du, and Martin D. F. Wong. A polynomial time exact algorithm for overlay-resist self-aligned double patterning (SADP) layout decomposition. 32(8):1228–1239, August 2013.

[76] Jingyu Xu, Subarna Sinha, and Charles C. Chiang. Accurate detection for process-hotspots with vias and incomplete specification. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 839–846, 2007.

[77] Yue Xu and Chris Chu. A matching based decomposer for double patterning lithography. In *ACM International Symposium on Physical Design (ISPD)*, pages 121–126, 2010.

[78] Jae-Seok Yang, Katrina Lu, Minsik Cho, Kun Yuan, and David Z. Pan. A new graph-theoretic, multi-objective layout decomposition framework for double patterning lithography. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 637–644, 2010.

[79] Hailong Yao, Subarna Sinha, Charles Chiang, Xianlong Hong, and Yici Cai. Efficient process-hotspot detection using range pattern matching. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 625–632, 2006.

[80] Bei Yu, Jhih-Rong Gao, Duo Ding, Yongchan Ban, Jae seok Yang, Kun Yuan, Minsik Cho, and Pan David Z. Pan. Dealing with IC manufacturability in extreme scaling. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 240–242, 2012.

[81] Bei Yu, Xiaoqing Xu, Jhih-Rong Gao, and David Z. Pan. Methodology for standard cell compliance and detailed placement for triple patterning lithography. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 349–356, 2013.

[82] Bei Yu, Kun Yuan, Jhih-Rong Gao, and David Z. Pan. E-BLOW: E-beam lithography overlapping aware stencil planning for MCC system. In *ACM/IEEE Design Automation Conference (DAC)*, pages 1–7, 2013.

[83] Peng Yu and David Z. Pan. TIP-OPC: a new topological invariant paradigm for pixel based optical proximity correction. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 847–853, 2007.

[84] Peng Yu, Sean X. Shi, and David Z. Pan. Process variation aware OPC with variational lithography modeling. In *ACM/IEEE Design Automation Conference (DAC)*, pages 785–790, 2006.

[85] Peng Yu, Sean X. Shi, and David Z. Pan. True process variation aware optical proximity correction with variational lithography modeling and model calibration. *Journal of Microlithography, Microfabrication and Microsystems*, 6(3):031004, September 2007.

[86] Yen-Ting Yu, Ya-Chung Chan, Subarna Sinha, Iris Hui-Ru Jiang, and Charles Chiang. Accurate process-hotspot detection using critical design rule extraction. In *ACM/IEEE Design Automation Conference (DAC)*, pages 1167–1172, 2012.

[87] Yen-Ting Yu, Geng-He Lin, Iris Hui-Ru Jiang, and Charles Chiang. Machine-learning-based hotspot detection using topological classification and critical feature extraction. In *ACM/IEEE Design Automation Conference (DAC)*, pages 67:1–67:6, 2013.

[88] Kun Yuan, Katrina Lu, and David Z. Pan. Double patterning lithography friendly detailed routing with redundant via consideration. In *ACM/IEEE Design Automation Conference (DAC)*, pages 63–66, July 2009.

[89] Kun Yuan and David Z. Pan. WISDOM: Wire spreading enhanced decomposition of masks in double patterning lithography. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 32 –38, 2010.

[90] Hongbo Zhang, Yuelin Du, M. D. Wong, and Rasit Topaloglu. Self-aligned double patterning decomposition for overlay minimization and hot

spot detection. In *ACM/IEEE Design Automation Conference (DAC)*, pages 71–76, 2011.

[91] Jinyu Zhang, Wei Xiong, Yan Wang, and Zhiping Yu. A highly efficient optimization algorithm for pixel manipulation in inverse lithography technique. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 480–487, November 2008.

[92] Jinyu Zhang, Wei Xiong, Yan Wang, Zhiping Yu, and Min-Chun Tsai. A robust pixel-based RET optimization algorithm independent of initial conditions. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 645–650, January 2010.

[93] Xin Zhao and Chris Chu. Line search-based inverse lithography technique for mask design. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 2012, January 2012.

Vita

Jhih-Rong Gao received the Bachelor and Master of Science degree in Computer Science from the National Tsing Hua University in 2005 and 2007, respectively. After graduation, she joined Synopsys in Taiwan as an R&D Engineer for two years. In 2009, she joined UT Design Automation Lab and started her Ph.D. study with Prof. David Z. Pan in the Electrical & Computer Engineering department in the University of Texas at Austin. She had interned with Synopsys in summer 2011, and Intel in summer 2012.

Jhih-Rong's main research involves lithography aware design automation and design for manufacturability. The impacts of Jhih-Rong's works are reconized with Best Paper Award nomination at IEEE/ACM Design Automation Conference 2014, SPIE BACUS Photomask scholarship in 2013, IEEE CEDA awards in the CAD contest at ICCAD 2012 and 2013. She has served as a reviewer for technical journals including IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems (TCAD) and IEEE Transactions on Very Large Scale Integration Systems (TVLSI).

Permanent address: jrgao@utexas.edu

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.