

Concurrent Guiding Template Assignment and Redundant Via Insertion for DSA-MP Hybrid Lithography

Jiaojiao Ou
ECE Department
Univ. of Texas at Austin
jiaojiao@cerc.utexas.edu

Bei Yu
CSE Department
Chinese Univ. of Hong Kong
byu@cse.cuhk.edu.hk

David Z. Pan
ECE Department
Univ. of Texas at Austin
dpan@ece.utexas.edu

ABSTRACT

Directed Self-Assembly (DSA) is a very promising emerging lithography for 7nm and beyond, where a coarse guiding template produced by conventional optical lithography can “magically” generate fine-pitch vias/contacts through self-assembly process. A key challenge for DSA-friendly layout is the guiding template assignment to cover all vias under consideration. Meanwhile, redundant via insertion has been widely adopted to improve yield and reliability of the circuit. In this paper, we propose a comprehensive framework for concurrent DSA guiding template assignment and redundant via insertion with consideration of multiple patterning (MP) in guiding template generation. We first formulate the problem as an integer linear programming (ILP), and then propose a novel approximation algorithm to achieve good performance and runtime trade-off. The experimental results demonstrate the effectiveness of the proposed algorithms. To our best knowledge, this is the first work in concurrent guiding template assignment and redundant via insertion for DSA-MP hybrid lithography.

CCS Concepts

•Hardware → VLSI design manufacturing considerations;

Keywords

Directed Self-Assembly, Multiple Patterning Lithography, Redundant Via Insertion, Guiding Template Assignment

1. INTRODUCTION

As a promising next generation lithography, Directed Self-Assembly (DSA) is gaining interests from both industry and academia, due to its ability to improve the contact/via pitch, as well as its low cost and high throughput [1–3]. A typical DSA process is depicted in Fig. 1, where at first some guiding templates are generated through conventional 193i lithography system. Then block copolymer (BCP) is filled in the guiding templates, followed by block copolymer annealing

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ISPD’16, April 03–06, 2016, Santa Rosa, CA, USA

© 2016 ACM. ISBN 978-1-4503-4039-7/16/04...\$15.00

DOI: <http://dx.doi.org/10.1145/2872334.2872352>

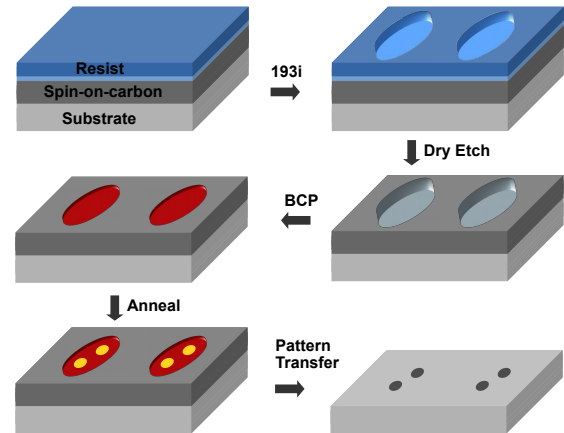


Figure 1: DSA process flow: 193i lithography and etch define the guiding template, the final DSA patterns are generated inside the template.

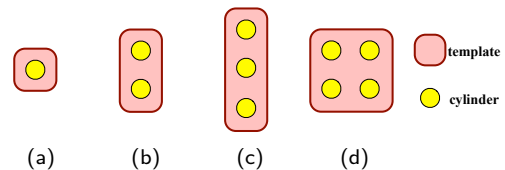


Figure 2: Robust DSA patterns: (a) Singlet. (b) Doublet. (c) Triplet. (d) Quadruplet.

to form cylinders. Since the guiding templates are printed by traditional 193i lithography, the cylinder pitch variation and placement error will deteriorate with the increase of template complexity. Therefore, in order to get better variation control, some regular shaped DSA guiding templates are preferred. Fig. 2 lists some typical regular guiding template examples, which are reported to have less pitch variations and better manufacturability [4]. These guiding templates are singlets, doublets, triplets and quadruplets, which are all in regular shapes.

In recent years, DSA lithography is an emerging candidate for the printing of contact/via layers in 7nm technology node and below [5]. There has been significant improvement made on the manufacturing, modeling and simulation of DSA, especially for grapho-epitaxy DSA [6, 7]. On the other hand, multiple patterning is often required for the manufacturing of the dense distributed contacts. It is reported that the application of DSA can also help to reduce the number of mask when contacts/vias are grouped in the same DSA guiding

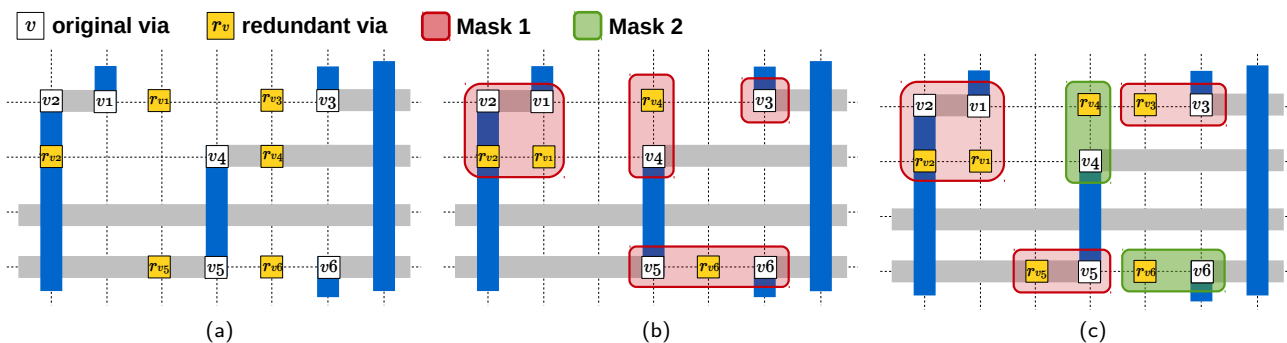


Figure 3: (a) Redundant via insertion result without consideration of DSA pattern compatibility. (b) Concurrent redundant via insertion (RVI) and GTA, insertion rate is reduced due to DSA design rule restrictions. (c) Concurrent GTA and RVI for DSA with double patterning so that the via insertion rate is improved.

template if their distances are within the optical resolution limit [8].

Much previous research has been proposed on contact layer DSA aware design (e.g. [9–11]). There are a lot of work focusing on DSA guiding template assignment (GTA) to minimize manufacturing variations as well. Du et al. [12] proposed a SAT algorithm and bounded approximation algorithm to explore the DSA aware contact layer optimization for 1D standard cell library. Xiao et al. [13] studied the DSA template determination and cut redistribution for cut mask in 1D gridded design, while Ou et al. [14] extended this problem by formulating it to an ILP problem and proposed another heuristic method to solve this problem. Badr et al. [15, 16] proposed a set of solutions to resolve the GTA and MP decomposition problem for contacts and vias.

For a long time, Redundant Via Insertion (RVI) has been heavily utilized in industry to improve circuit yield and reliability. Fig. 3(a) shows an example of RVI, where an extra via near the original single via is inserted as the redundant via. RVI has been well studied in routing and post-routing stages, where the major target is to maximize the via insertion rate [17–22]. The redundant via insertion problem has been formulated as a maximum independent set problem in post-routing stage by Lee et al. [17]. In [20], this problem has been formulated as a bipartite matching problem.

For 14nm technology nodes and beyond, it becomes quite difficult to print the lower via layer with 193i lithography single patterning as the density of vias dramatically increases. Thus multiple patterning (MP), such as Double Patterning (DP) and Triple Patterning (TP), is required for via layer fabrication. However, if we use DSA to print the via layer, and continue to use the conventional RVI method without DSA pattern constraints, it is very likely that the dense randomly distributed vias can make the grouping of vias to the regular shaped DSA guiding template very difficult without violating any design rules. Therefore, it is necessary to consider the DSA guiding template patterns and distributions of redundant vias at the insertion stage, to make it compatible when grouping vias to different guiding patterns, as shown in Fig. 3(b). We also notice that as more constraints have been introduced to the DSA redundant via insertion, the insertion rate may be impaired under very strict design rules, v_3 and v_5 are dead vias in Fig. 3(b). In order to maintain

a relatively high insertion rate, multiple patterning for DSA guiding templates and redundant via insertion are considered simultaneously in this work. Fig. 3(c) gives an example of redundant via insertion and DSA GTA under DSA-double patterning hybrid lithography, which could reach 100% insertion rate. In addition, it is also necessary to cover as many vias as possible by using the DSA patterns. Otherwise, we need to use other technique to print the vias that can not be patterned by DSA, which is not cost-effective. But it is still unknown how to solve the problem in DSA-MP hybrid lithography.

In this paper, we investigate the RVI and GTA for DSA lithography with multiple patterning. Our contributions of this paper can be summarized as follows:

- We consider DSA guiding template assignment together with multiple patterning for redundant via insertion.
- We model this problem as a constrained maximum weighted matching for bipartite graph problem, and propose an integer linear programming (ILP) formulation to search for the optimal solutions.
- To improve scalability, we further propose a linear programming (LP) based approximation algorithm to improve the runtime without much loss of solution quality.
- Our experimental results are promising in terms of redundant via insertion rate, total number of patterned vias, and runtime.

The rest of the paper is organized as follows: Section 2 introduces the background of redundant via insertion and the problem formulation. Section 3 proposes the constrained bipartite matching algorithm and general ILP formulation. Section 4 develops an approximation algorithm for further speed-up. Section 5 presents the experimental results, then followed by conclusion in Section 6.

2. PRELIMINARIES

In this section, we first provide some preliminaries on redundant via insertion and DSA guiding template candidates construction. Then we will give the problem formulation.

2.1 Redundant Via Insertion

A redundant via can be inserted in one of the adjacent

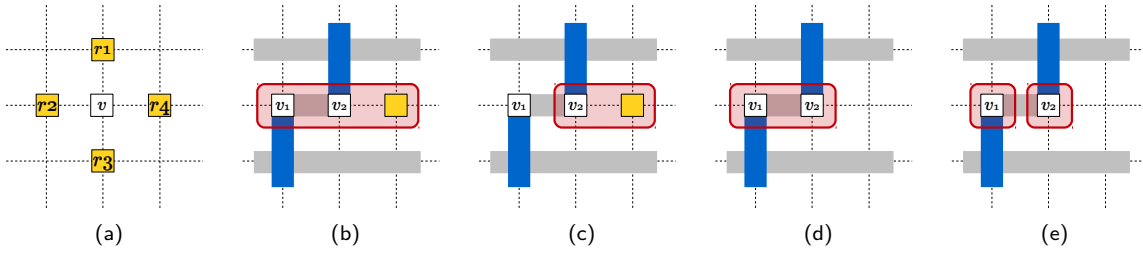


Figure 4: (a) Redundant via positions. (b)-(c) Feasible DSA guiding template candidates that have redundant vias. (d)-(e) DSA guiding template candidates without redundant via.

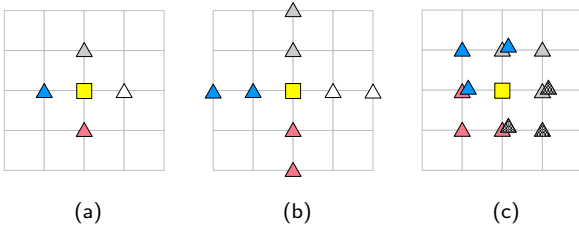


Figure 5: (a) Searched space for doublets, (b) triplets, (c) quadruplets.

four positions of the original single via, as shown in Fig. 4(a). The redundant via is feasible if it does not violate any design rules or overlap with other vias, otherwise it is regarded as infeasible. Generally speaking, if we consider two metal layers in the design, the adjacent position of a single via is feasible to insert redundant via when it satisfies the following conditions: it does not overlap with critical area, it is not occupied by any metal layer, or it is occupied by only one metal layer, and the metal layer is on the same net with the single via. For via that has spaces to insert a redundant via is referred as an *alive via*. While a via is called as a *dead via* if there is no place to insert a redundant via. The conventional evaluation metric for redundant via insertion is the insertion rate (IR), which is the ratio of inserted redundant via number over the original single via number. The ideal insertion is 100%, i.e. each single via has a redundant via. The insertion of redundant via can be performed in the routing stage or post-routing stage. While in this work, we consider the redundant via insertion in the post-routing stage for two routing layers to improve circuit reliability and yield.

2.2 Guiding Template Assignment

Given a post-routing layout and a set of DSA guiding templates, we need to find all the DSA guiding template assignment for all the single vias and feasible redundant vias. For simplicity, it is assumed that the original single via and its redundant via should be included in the same template. We also notice that the insertion rate can not be improved if the template does not include any redundant via, as shown in Fig. 4(d) and Fig. 4(e), but this kind of DSA guiding template assignment could help improve the number of covered vias for some special case, thus we need to include these templates. Figs. 4(b)-(c) illustrate the DSA guiding template candidates which have redundant vias, Figs. 4(d)-(e) illustrate the template candidates with only original single vias. In order to generalize this problem, it is assumed that the metal lines and vias are on grids, and redundant via can be placed at adjacent four grids.

To find DSA guiding template candidates for the entire

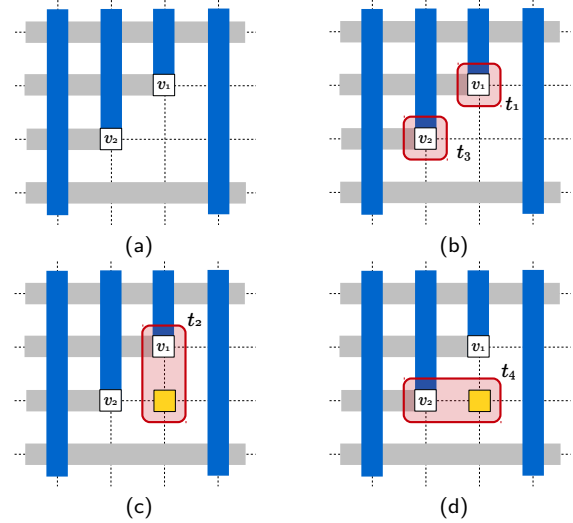


Figure 6: (a) Original layout. (b)-(d) All possible GTA result for the layout.

layout, we iterate all the single vias and verify the adjacent grids of each single via to determine whether the grids can be combined with current single via to form a valid DSA template pattern. The search method for different DSA pattern is as follows. We first explore the adjacent four grids of one single via, as illustrated in Fig. 5(a), we verify the grids one by one to check whether it is feasible to insert a redundant via, or it is another single via, so that it can be combined with the original via to form a doublet DSA pattern. We then explore the adjacent two grids of the single via, and there are four directions that should be checked, as shown in Fig. 5(b), each color indicates the search space in the four directions. The adjacent two grids can be combined with the original via to form a triplet DSA guiding pattern if one of the following conditions can be satisfied:

- The nearest grid contains another single via and the next grid is feasible to insert a redundant via for the second single via;
- The nearest grid is feasible to insert a redundant via for the current single via and the next grid contains another single via;
- Both grids contain single vias.

For the quadruplet DSA guiding pattern, it is a little different from the previous two cases. The search space is the adjacent three grids, and there are four cases to be checked as shown in different colors in Fig. 5(c). The single via and its three adjacent positions can be grouped as a quadruplet DSA guiding template if the adjacent grids have at least one single via and other grids are the feasible positions to

insert redundant vias for the single vias. Since two vias can be grouped together in one guiding templates, the via pitch can be reduced compared to traditional lithography.

Fig. 6 shows all the DSA guiding template assignment for the example layout. v_1 and v_2 are the original single vias, and there are four possible groups of the single via and redundant vias of DSA guiding template: t_1, t_2, t_3, t_4 . The DSA guiding templates and vias are stored as rectangles in a R-tree structure for later use.

2.3 Problem Formulation

The primary target of redundant via insertion is to maximize the number of inserted redundant vias in order to improve the yield and reliability of chips. we also consider the number of vias that can be patterned by DSA guiding template for manufacturing compatibility. The application of multiple patterning can help to improve the insertion rate. Thus the guiding template assignment for redundant via insertion of DSA-MP hybrid lithography (GTAR) problem is defined as follows:

Problem 1 (GTAR). *Given a post-routing layout with metal layers and via layer, a predefined DSA guiding template set, and the number of masks for the via layer, our objective is to maximize the redundant via insertion rate and the number of vias patterned by DSA guiding templates.*

3. ALGORITHM

In this section, we model this problem as a constrained weighted matching of bipartite graph problem. Then an ILP formulation for the constrained bipartite matching algorithm is proposed to search for the solutions.

3.1 Constrained Bipartite Graph Matching

The partition in the constrained bipartite graph is the vertex set for single vias and DSA guiding template candidates in this work, which is different from traditional maximum bipartite matching methods mentioned in [17, 20], where the partition is the set of single vias and redundant vias in the graph. An edge is added between the original via and the DSA guiding template if the via is covered by the template, as shown in Fig. 7(a), which is the bipartite graph for the example layout in Fig. 6. This bipartite graph is constrained for the reason that the edges are divided into different edge set, and the maximum number of edges for each set is limited. We assume that each edge set has been assigned with a certain color, the edge set constraint is referred as the color constraint in this paper. For each template candidate, if there exists another template that has overlap conflict with it, the corresponding edges are assigned to the same color (edge set), as shown in Fig. 7(b), in which template t_1 overlaps with t_2 , so edges connected with t_1 and t_2 are assigned to an edge set with the same color, and at most one edge can be selected from this color. The color constraint assignment is similar for other overlapped templates, as shown in Figs. 7(c)–7(d). If the template has other design rule violation, the corresponding edges are assigned to another color, as shown in Figs. 7(e)–7(f), where t_1 and t_3 , t_1 and t_4 , t_2 and t_3 , have minimum distance violation between the DSA templates. For the ordinary matching, as there is no priority for the edges, it is possible that most of the DSA templates of the matching result does not contain any redundant via, or most of the vias are not patterned by the DSA pattern. In

Table 1: Notation

G	Constrained bipartite graph
V	Vertex in the bipartite graph
V_S	Vertex set of single via
V_T	Vertex set of DSA guiding template assignment
E_i	Edge set with color i
C_i	Color constraint for edge set E_i
V_{TW}	Vertex set of DSA guiding template assignment for MP
e_i	Edge i
e_{ij}	Multiplied edge of e_i on mask j
E_{ij}	Multiplied edge set of E_i on mask j
EO_{ij}	Edge set that has overlap conflict with edge e_{ij}
EV_{ij}	Edge set that has other violations with edge e_{ij}
EV_k	Edge set for edges connected with vertex k
W	Multiple patterning number
w_i	Maximum number of edges for C_i
$x_{e_{ij}}$	0/1 variable to indicate the value of edge e_{ij}
$\alpha \setminus \beta$	Weight of DSA template candidate with and without redundant via
$E_w \setminus E_{wo}$	Edge set with and without redundant via

order to obtain a balance between the insertion rate and the number of covered single vias, we can assign different weight to these edges: we assign a large weight α to the edges connecting with template which includes redundant via, and a relatively small weight β to other edges. Thus, we are able to maintain a high insertion rate and a high coverage of vias at the same time.

For simplicity, we first consider the constrained bipartite matching without multiple patterning. Some notations are illustrated in Table 1. The constrained bipartite graph is constructed for single patterning, as shown in Fig. 7.

Problem 2 (Constrained Weighted Matching). *Given a weighted bipartite graph $G = (V, E)$ with partition $V = V_S \cup V_T$, the edge set is partitioned to m set $E_1 \cup E_2 \cup \dots \cup E_m$, each set has color C_i . Find the maximum weighted matching M such that there are at most one edge of color C_i , i.e. $|M \cap E_i| \leq 1, \forall i \in [m]$.*

If multiple patterning is considered simultaneously, for DSA guiding templates that have other design rule violations, they can be assigned to different masks to increase the insertion rate. The vertex set for templates V_T and edge set E are multiplied to indicate the masks they are assigned to. Fig. 8 shows the bipartite graph and different color constraints of edge set for double patterning. It is noted that the number of edges and vertexes for templates has been doubled, the color constraints assignment is similar to single patterning. It is easy to extend the bipartite graph and color constraints construction to other multiple patterning in the similar way.

Problem 3 (Constrained Weighted Matching for MP).

Assume it is a W multiple patterning. Given a bipartite graph $G = (V_W, E_W)$ with partition $V_W = V_S \cup V_{TW}$, the number of vertexes and edges in V_{TW} and E_W are multiplied: $|V_{TW}| = W \times |V_T|$, $|E_W| = W \times |E|$. Edge set is partitioned to m set: $E_W = E_1 \cup E_2 \cup \dots \cup E_m$, each subset has color C_i . Find the maximum weighted matching M such that there are at most one edge for overlapping color constraint C_i , i.e. $|M \cap E_i| \leq 1, i \in (1, 2, \dots, m)$, and at most two edges for other color constraint C_i , i.e. $|M \cap E_i| \leq 2, i \in (1, 2, \dots, m)$.

3.2 ILP Formulation

Since maximum weighted constrained matching in bipartite graph is NP-hard [23, 24], we provide an ILP formulation to search for optimal solution.

Let N_v indicate the number of original single via, N_t indicate the number of templates. Let e_{ij} denote edge ij , the

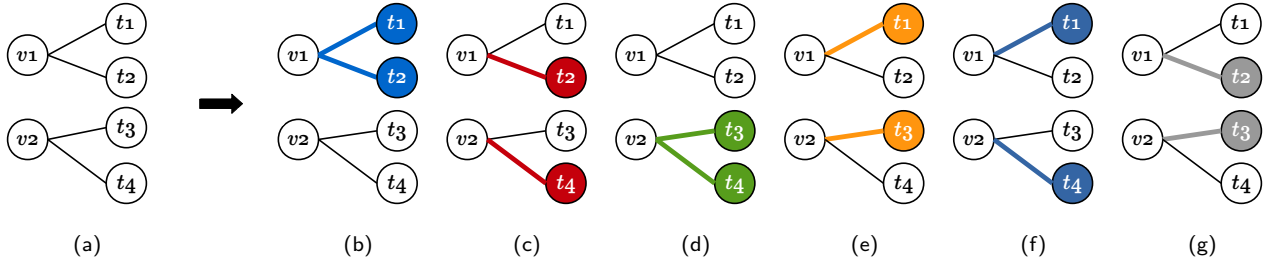


Figure 7: Constrained bipartite graph for layout in Fig. 6. (a) Initial bipartite graph. (b)-(d) Color constraints: edge set for edges connected with overlapped DSA guiding template candidates. (e)-(g) edge set for edges that connected with DSA guiding template candidates which have other design rule violations.

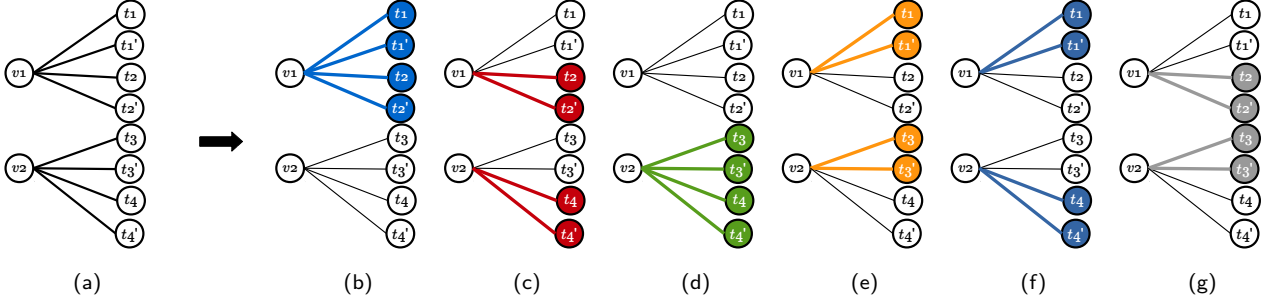


Figure 8: Constrained bipartite graph for layout in Fig. 6 with double patterning. (a) Initial bipartite graph. (b)-(d) Color constraints: edge set for edges connected with overlapped DSA guiding template candidates. (e)-(g) edge set for edges that connected with DSA guiding template candidates which have other design rule violations.

index ij means the primary edge e_i is on mask j . Let $x_{e_{ij}}$ ($e_{ij} \in E$) indicate the binary value for this edge. Other notations are defined in Table 1. The constrained maximum weighted matching for multiple patterning can be formulated as follows:

$$\begin{aligned}
 & \text{maximize } \alpha \sum_{e_{ij} \in E_w} x_{e_{ij}} + \beta \sum_{e_{ij} \in E_{wo}} x_{e_{ij}} & (1) \\
 & \text{s.t.} \\
 & \sum_{e_{ij} \in E_{V_k}} x_{e_{ij}} \leq 1, \quad \forall k \in (0, \dots, N_v) \\
 & \sum_{e_{ij} \in E_{V_h}} x_{e_{ij}} \leq 1, \quad \forall h \in (0, \dots, N_t) \\
 & x_{e_{ij}} + x_{e_{\tilde{j}\tilde{j}}} \leq 1, \quad \forall e_{ij} \in E, \forall e_{\tilde{j}\tilde{j}} \in EO_{ij} \\
 & x_{e_{ij}} + x_{e_{\tilde{j}\tilde{j}}} \leq 2, \quad \forall e_{ij} \in E, \forall e_{\tilde{j}\tilde{j}} \in EV_{ij}, \tilde{j} \neq j \\
 & x_{e_{ij}} + x_{e_{\tilde{j}\tilde{j}}} \leq 1, \quad \forall e_{ij} \in E, \forall e_{\tilde{j}\tilde{j}} \in EV_{ij}, \tilde{j} = j
 \end{aligned}$$

The target is to maximize the number of edges in the matching. The first constraint indicates that there is at most one edge that can be selected for each vertex, which corresponds to the matching. The second constraint indicates that if two edges, e_{ij} and $e_{\tilde{j}\tilde{j}}$, have overlap conflict, then at most one edge can be selected. $e_{\tilde{j}\tilde{j}}$ belongs to the edge set EO_{ij} which contains edges that have overlap conflict with edge e_{ij} . The third constraint indicates that for any two templates that have other design rule violations, they can be selected at the same time if they are assigned to different masks ($j \neq \tilde{j}$), here EV_{ij} indicates edge set which contains edges that have design rule violation with edge e_{ij} . However, they are not allowed on the same mask, as illustrated in the last constraint.

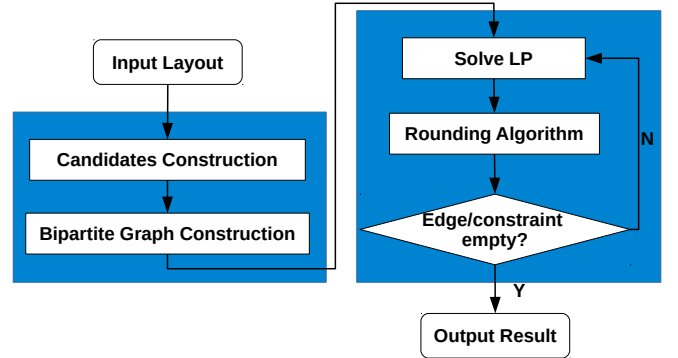


Figure 9: Basic flow for LP approximation algorithm.

We notice that edges which connecting with the same template or vias are also overlapping with each other, which is the same as the third constraint. Thus the simplified ILP formulation is as follows:

$$\begin{aligned}
 & \text{maximize } \alpha \sum_{e_{ij} \in E_w} x_{e_{ij}} + \beta \sum_{e_{ij} \in E_{wo}} x_{e_{ij}} & (2) \\
 & \text{s.t.} \\
 & x_{e_{ij}} + x_{e_{\tilde{j}\tilde{j}}} \leq 1, \quad \forall e_{ij} \in E, \forall e_{\tilde{j}\tilde{j}} \in EO_{ij} \\
 & x_{e_{ij}} + x_{e_{\tilde{j}\tilde{j}}} \leq 2, \quad \forall e_{ij} \in E, \forall e_{\tilde{j}\tilde{j}} \in EV_{ij}, \tilde{j} \neq j \\
 & x_{e_{ij}} + x_{e_{\tilde{j}\tilde{j}}} \leq 1, \quad \forall e_{ij} \in E, \forall e_{\tilde{j}\tilde{j}} \in EV_{ij}, \tilde{j} = j
 \end{aligned}$$

4. APPROXIMATION ALGORITHM

Since ILP is an NP-hard problem, the runtime penalty is quite large with the increase of problem size, especially

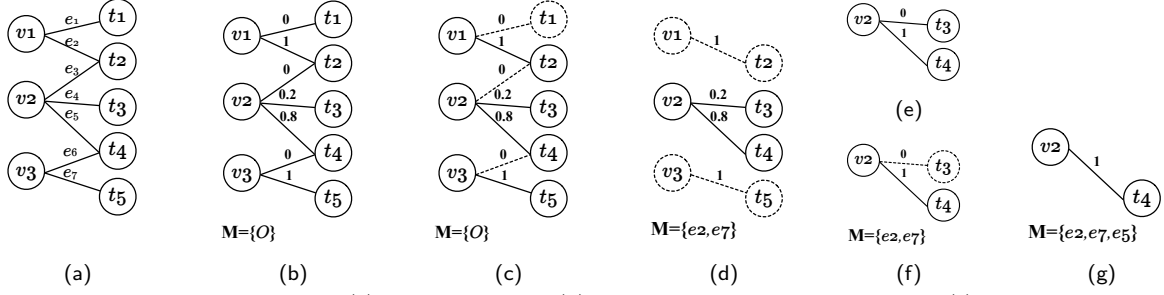


Figure 10: LP relaxation and rounding. (a) Bipartite graph. (b) Initial solution of LP formulation. (c) Remove edges with 0 value, then remove corresponding vertexes with zero degree. Edges and vertexes to be removed are marked as dashed lines. (d) Add edges with value 1 to the final matching M , then remove these vertexes and edges. (e) For tight vertex, round its edge to 1 if the edge value is greater than $\frac{1}{2}$, otherwise round the edge value to 0. (f)-(g) Remove edges with 0 value and vertexes with zero degree, add edges with value 1 to the final matching M .

for multiple patterning where the number of edges has been multiplied. We apply an LP approximation algorithm to solve the constrained maximum weighted matching in bipartite graph problem more efficiently without much loss of solution quality. Our approximation algorithm is inspired by paper [23], which talks about the constrained maximum weighted matching in bipartite graphs. The basic flow of our approximation method is shown in Fig. 9. Our approximation flow consists of two steps: LP relaxation and rounding. Some of the notations used in the algorithm are also defined in Table 1. For generality, we use e to denote an edge, $v \setminus t$ to denote a vertex for via\template in the bipartite graph, and $e = (v, t)$.

For the LP relaxation, we relax the ILP formulation into an LP by replacing the binary variable of edge e with continuous variable $x_e \in [0, 1]$, to obtain a feasible solution x for the current LP. Then we first remove edge e from the edge set E if its value is zero, i.e. $x_e = 0$, and remove the vertex if its degree value is zero, i.e. $deg(v) = 0$ or $deg(t) = 0$. We add the edge to our final bipartite matching M if $x_e = 1$, i.e. $M = M \cup e$. Then we need to update the vertex set to remove the edge $e = (v, t)$ when $x_e = 1$, i.e. $V = V \setminus \{v, t\}$; Then update the maximum number of edges for different color constraints: if the value of the edge $x_e = 1$, all its corresponding color constraints should reduce the maximum number by one, if $e \in E_i$, then $w_i = w_i - 1$. Generally speaking, as the initial value of w is 1 or 2 in the ILP formulation, w usually decreases to zero very quickly. If the maximum number of edges for color constraint is zero, i.e. $w = 0$, we then remove this color constraint and all the edges connected with this constraint. The removal of these edges can introduce more zero degree vertexes, we then delete the newly generated zero degree vertexes.

For the rounding step, we round some edges to 1 and others to 0: If there exists a tight vertex $v \in V$ such that $deg(v) = 2$, assume the two edges connected with vertex v are e_1 and e_2 , and $e_1 = (v, t_1), e_2 = (v, t_2)$. If $x_{e_1} \geq \frac{1}{2}$, we round x_{e_1} to 1, and add e_1 to our final matching set M . Then round x_{e_2} and all the other edges connected with t_1 to zero. Then update the maximum number of edges for the color constraint: if $e_1 \in E_i$, thus $w_i = w_i - x_{e_1} - \lambda(1 - x_{e_1})$, $\lambda \in [0, 1]$. And vice versa. We should note that λ can be different in different iterations. We then update the vertex set to remove the rounded edges and vertexes from bipartite graph. Then iterate the algorithm from the LP relaxation. The details of relaxation and rounding algorithm is shown

in Algorithm 1,

Algorithm 1 Approximation for Constrained Weighted Matching in Bipartite Graph

Input: Final matching $M = \emptyset$

- 1: **while** $C \neq \emptyset$ or $E \neq \emptyset$ **do**
- 2: Obtain basic feasible solution x for current LP;
- 3: Remove edge e from the graph when $x_e = 0$;
- 4: Remove node when $deg(v) = 0$;
- 5: **if** $x_e = 1$ **then**
- 6: add edge $e = (v, t)$ to M , $M = M \cup e$;
- 7: **end if**
- 8: Iterate from 1;
- 9: Remove corresponding nodes $V = V \setminus \{v, t\}$;
- 10: Update constraint, $w = w - 1$ if $x_e = 1$;
- 11: **if** $w = 0$ and $w \in C_i$ **then**
- 12: update constraints set $C = C \setminus C_i$;
- 13: **end if**
- 14: Update edge set $E = E \setminus \{e : e \in E_i\}$;
- 15: Remove nodes with zero degree;
- 16: **(Rounding)**
- 17: **if** $\exists v$ such that $deg(v) = 2$, assume $e_1 = (v, t_1), e_2 = (v, t_2)$ **then**
- 18: **if** x_{e_1} or $x_{e_2} \geq 1/2$ **then**
- 19: Rounding x_{e_1} or x_{e_2} to 1, add it to M ;
- 20: Rounding x_{e_2} or x_{e_1} and all other edges connected with t_1 or t_2 to zero;
- 21: If e_1 or $e_2 \in E_i$, then $w_i = w_i - x_{e_1} - \lambda(1 - x_{e_1})$ or $w_i = w_i - x_{e_2} - \lambda(1 - x_{e_2})$;
- 22: Remove (v, t_1) or (v, t_2) and all rounded edges from the graph;
- 23: **end if**
- 24: **end if**
- 25: **end while**
- 26: Return M ;

Fig. 10 gives a simple example for the relaxation and rounding algorithm, in which Fig. 10(a) shows the original bipartite graph. We first obtain the solution of LP, the value for each edge is shown in Fig. 10(b). Then remove edges e_1, e_3 , and e_6 as their values are zero, as shown in Fig. 10(c). Then remove vertex t_1 as its degree is zero. Because the values of edges e_2, e_7 equal to one, we then add them to the matching M , $M = \{e_2, e_7\}$, and remove corresponding edges and vertexes, as shown in Fig. 10(d). For

edges e_4 and e_5 that connect with tight vertex v_2 , round the value of edge e_5 to one as its value is greater than $\frac{1}{2}$, and round e_4 to zero as its value is smaller than $\frac{1}{2}$. Remove edge e_4 , then remove vertex t_3 as its degree is zero. At last add edge e_5 to matching M , $M = \{e_2, e_7, e_5\}$. The process is done when the edge set is empty.

It should be noted that there might be some violations of the color constraints during the updating of maximum number of edges. There may be more color violation for smaller λ value, but better objective performance. Thus in order to prevent the color constraints violations, we can set $\lambda = 1$. Once an edge is rounded, the maximum number of edges decreases by one.

5. EXPERIMENTAL RESULT

All algorithms in this work are implemented in C++, and they are performed on a Linux workstation with Intel Core i7 3.4GHz CPU and 32GB memory. CBC [25] is used as the solver for ILP and LP. Benchmarks used in the experiments are based on the OpenSPARC T1 design [26]. Different modules in OpenSPARC T1 are synthesized with Design Compiler [27]. We then place and route the benchmarks with NanGate 45nm open cell library [28] by using Cadence SOC encounter [29], where the utilization rate is set to 0.7. The weight ratio ($\frac{\alpha}{\beta}$) of DSA template candidates with and without redundant via is set to 250. In order to ensure that no violation is generated during the approximation, the coefficient λ used in the updating of maximum number of edges for color constraints is set to 1. Metal two and metal three layers are used for the redundant via insertion, as they are the most dense metal layers among all the routing layers. We first convert the GDSII layout into grid, and then search for DSA guiding template candidates based on the grids. We assume that there are two units for one metal pitch, and the minimum distance requirement between different DSA guiding template is two metal pitch. Our algorithm is adaptable to more advanced technology node as it is graph based and can be applied to different layout.

In order to compare the performance of our algorithm for DSA and multiple patterning aware redundant via insertion, we implement the DSA aware redundant via insertion with single patterning (SP), double patterning (DP), and triple patterning (TP) to evaluate the improvement of inserted redundant via (insertion rate). We implement the ILP for single patterning (SP-ILP), double patterning (DP-ILP) and triple patterning (TP-ILP), and corresponding approximation algorithms for DP and TP (DP-Ap and TP-Ap). We also implement the ILP formulation for un-constrained redundant via mentioned in [21] (Un-constrained) as comparison, since this algorithm inserts the redundant via without any DSA pattern constraints, it can reach the ideal maximum insertion rate for the layout, but with a lot of vias that can not be patterned by DSA. We use the insertion rate of un-constrained RVI as the ideal target for DSA aware RVI. We assume the design rule is the same for all implementations. In order to evaluate the performance of these methods, we define the coverage rate (CR) to indicate the ratio of patterned single vias and original single vias, we would like to achieve a high insertion rate IR and high coverage rate CR .

The insertion rate (IR) is compared in Table 2. The second column is the total number of original vias for each benchmark. The third column shows the insertion rate for

Table 3: Coverage Rate (CR) Comparison

Bench	#Vias	SP-ILP	DP-ILP	DP-Ap	TP-ILP	TP-Ap
efc	4983	76.33	100	98.63	100	99.53
ecc	5523	80.10	100	98.55	100	99.67
ffu	7026	78.49	100	98.83	100	99.45
alu	7046	74.65	100	98.14	100	99.16
byp	28847	75.14	N/A	97.29	N/A	98.89
mul	62989	70.23	N/A	96.39	N/A	98.31
Avg.	19402	75.82	N/A	97.97	N/A	99.17

Table 4: Runtime Comparison (s)

Bench	#Vias	SP-ILP	DP-ILP	DP-Ap	TP-ILP	TP-Ap
efc	4983	0.85	68.85	3.36	85.07	7.38
ecc	5523	0.84	56.26	3.80	119.43	7.93
ffu	7026	1.17	88.64	4.99	98.22	12.02
alu	7046	1.33	107.34	5.40	399.38	11.55
byp	28847	6.31	N/A	41.41	N/A	136.69
mul	62989	30.98	N/A	417.76	N/A	1613.23
Avg.	19402	6.91	N/A	79.45	N/A	298.13

un-constrained RVI, it represents the ideal maximum insertion rate for each benchmark without DSA pattern constraints, but it contains many vias that can not be patterned by DSA. When DSA pattern is considered in the redundant via insertion process with single patterning (SP), the insertion rate decreases about 24.00% on average due to the design rule violations between DSA guiding template when compared to the un-constrained method with ideal insertion rate. If double patterning process is considered simultaneously in the insertion process, since the violated templates can be assigned to different masks, its IR improves more than 20.00% based on available data when compared to the one with single patterning. When compared with the un-constrained method, the reduction of IR are less than 0.30% and 2.10% for solutions of DP-ILP and DP-Ap. For triple patterning process, when compared with the un-constrained method with ideal insertion rate, the reduction on the insertion rate are less than 0.05% and 0.86% for solutions of TP-ILP and TP-Ap. Therefore, the insertion rate of DSA-MP is quite comparable to un-constrained redundant via insertion, and the final insertion result is DSA compatible.

The coverage rate (CR) is shown in Table 3. As we can see from the table, the single patterning can cover most of the vias, but there are still 24% of vias that can not be patterned. For DP-ILP and TP-ILP, we can reach a hundred percent coverage. For DP-Ap and TP-Ap, the coverage rate can reach 97.97% and 99.17% on average.

The runtime is compared in Table 4. Here the runtime is the time consumption of CPU time to solve the problem, which excludes I/O processing time. The runtime of DP-ILP and TP-ILP increases dramatically because of more variables and constraints for large benchmarks, and the runtime of *byp* and *mul* is too long to be collected. Our approximation algorithm, DP-Ap and TP-Ap, are much faster. As the runtime of ILP for *byp* and *mul* is not available, we compare the result with largest available benchmark *alu*, DP-Ap is more than 20 \times faster than DP-ILP, and TP-Ap is more than 34 \times faster than TP-ILP.

6. CONCLUSION

Directed Self-Assembly is the potential candidate for the next generation lithography technique, which arouses a lot of interests in the DSA aware design and guiding template optimization. In this paper, we propose a general integer linear programming (ILP) formulation to solve the DSA aware

Table 2: Insertion Rate (IR) Comparison

Bench	#Vias	Un-constrained	SP-ILP	DP-ILP	DP-Ap	TP-ILP	TP-Ap
efc	4983	98.45	75.15	98.29	96.84	98.41	97.89
ecc	5523	99.02	78.68	98.80	97.44	98.98	98.62
ffu	7026	98.57	76.77	98.47	97.35	98.54	97.93
alu	7046	98.32	72.79	98.15	96.39	98.29	97.44
byp	28847	93.88	70.21	N/A	91.35	N/A	92.88
mul	62989	98.55	68.59	N/A	94.91	N/A	96.86
Avg.	19402	97.80	73.70	N/A	95.71	N/A	96.94

redundant via insertion with multiple patterning simultaneously. To improve scalability, we propose an approximation algorithm with LP relaxation and rounding to solve the problem efficiently. The experimental results demonstrate that our methods can insert redundant via with the consideration of DSA guiding template shapes without much loss of insertion rate. Our methods can be adaptive to different technology nodes as well.

Acknowledgment

This work is supported in part by National Science Foundation (NSF), Semiconductor Research Corporation (SRC), and CUHK Direct Grant for Research.

7. REFERENCES

- [1] Yuriko Seino, Hiroki Yonemitsu, Hironobu Sato, Masahiro Kanno, Hirokazu Kato, Katsutoshi Kobayashi, Ayako Kawanishi, Tsukasa Azuma, Makoto Muramatsu, Seiji Nagahara, Takahiro Kitano, and Takayuki Toshima. Contact hole shrink process using graphoepitaxial directed self-assembly lithography. *JM3*, 12(3), 2013.
- [2] Seong-Jun Jeong, Ju Young Kim, Bong Hoon Kim, Hyoung-Seok Moon, and Sang Ouk Kim. Directed self-assembly of block copolymer for next generation nanolithography. *Materials Today*, 16(12):468–476, 2013.
- [3] David Z Pan, Bei Yu, and J-R Gao. Design for manufacturing with emerging nanolithography. *IEEE TCAD*, 32(10):1453–1472, 2013.
- [4] H.-S. Philip Wong, Chris Bencher, He Yi, Xin-Yu Bao, and Li-Wen Chang. Block copolymer directed self-assembly enables sublithographic patterning for device fabrication. In *Proc. SPIE*, volume 8323, 2012.
- [5] Yuansheng Ma, Junjiang Lei, Juan Andres Torres, Le Hong, James Word, Germain Fenger, Alexander Tritchkov, George Lippincott, Rachit Gupta, Neal Lafferty, Yuan He, Joost Bekeart, and Geert Vanderberghe. Directed self-assembly (dsa) grapho-epitaxy template generation with immersion lithography. In *Proc. SPIE*, volume 9423, 2015.
- [6] Azat Latypov, Tamer H. Coskun, Grant Garner, Moshe Preil, Gerard Schmid, Ji Xu, and Yi Zou. Simulations of spatial DSA morphology, DSA-aware assist features and block copolymer-homopolymer blends. In *Proc. SPIE*, volume 9049, 2014.
- [7] Sander Wuister, Tamara Druzhinina, Davide Ambesi, Bart Laenens, Linda He Yi, and Jo Finders. Influence of litho patterning on DSA placement errors. In *Proc. SPIE*, volume 9049, 2014.
- [8] Yuansheng Ma, J. Andres Torres, Germain Fenger, Yuri Granik, Julien Ryckaert, Geert Vanderberghe, Joost Bekeart, and James Word. Challenges and opportunities in applying grapho-epitaxy DSA lithography to metal cut and contact/via applications. In *Proc. SPIE*, volume 9231, 2014.
- [9] He Yi, Xin-Yu Bao, Jie Zhang, Richard Tiberio, James Conway, Li-Wen Chang, Subhasish Mitra, and H.-S. Philip Wong. Contact-hole patterning for random logic circuit using block copolymer directed self-assembly. In *Proc. SPIE*, volume 8323, 2012.
- [10] Yuelin Du, Zigang Xiao, Martin D.F. Wong, He Yi, and H.-S. Philip Wong. DSA-aware detailed routing for via layer optimization. In *Proc. SPIE*, volume 9049, 2014.
- [11] Zigang Xiao, Yuelin Du, Haitong Tian, Martin D. F. Wong, He Yi, H-S Philip Wong, and Hongbo Zhang. Directed self-assembly (DSA) template pattern verification. In *Proc. DAC*, pages 55:1–55:6, 2014.
- [12] Yuelin Du, Daifeng Guo, Martin D. F. Wong, He Yi, H.-S. Philip Wong, Hongbo Zhang, and Qiang Ma. Block copolymer directed self-assembly (DSA) aware contact layer optimization for 10 nm 1D standard cell library. In *Proc. ICCAD*, pages 186–193, 2013.
- [13] Zigang Xiao, Yuelin Du, Martin D.F. Wong, and Hongbo Zhang. DSA template mask determination and cut redistribution for advanced 1D gridded design. In *Proc. SPIE*, volume 8880, 2013.
- [14] Jiaojiao Ou, Bei Yu, Jih-Rong Gao, and David Z. Pan. Directed self-assembly cut mask assignment for unidirectional design. *JM3*, 14(3), 2015.
- [15] Yasmine Badr, Andres Torres, and Puneet Gupta. Mask assignment and synthesis of DSA-MP hybrid lithography for sub-7nm contacts/vias. In *Proc. DAC*, pages 70:1–70:6, 2015.
- [16] Yasmine Badr, Juan Andres Torres, Yuansheng Ma, Joydeep Mitra, and Puneet Gupta. Incorporating DSA in multipatterning semiconductor manufacturing technologies. In *Proc. SPIE*, volume 9427, 2015.
- [17] Kuang-Yao Lee and Ting-Chi Wang. Post-routing redundant via insertion for yield/reliability improvement. In *Proc. ASPDAC*, pages 303–308, 2006.
- [18] Kuang-Yao Lee, Cheng-Kok Koh, Ting-Chi Wang, and Kai-Yuan Chao. Fast and optimal redundant via insertion. *IEEE TCAD*, 27(12):2197–2208, 2008.
- [19] Gang Xu, Li-Da Huang, David Z Pan, and Martin DF Wong. Redundant-via enhanced maze routing for yield improvement. In *Proc. ASPDAC*, pages 1148–1151, 2005.
- [20] Huang-Yu Chen, Mei-Fang Chiang, Yao-Wen Chang, Lumdo Chen, and Brian Han. Full-chip routing considering double-via insertion. *IEEE TCAD*, 27(5):844–857, 2008.
- [21] Kuang-Yao Lee, Cheng-Kok Koh, Ting-Chi Wang, and Kai-Yuan Chao. Optimal post-routing redundant via insertion. In *Proc. ISPD*, pages 111–117, 2008.
- [22] Jiwoo Pak, Bei Yu, and David Z. Pan. Electromigration-aware redundant via insertion. In *Proc. ASPDAC*, pages 544–549, 2015.
- [23] Monaldo Mastrolili and Georgios Stamoulis. Constrained matching problems in bipartite graphs. In *Proc. ISCO*, pages 344–355, 2014.
- [24] Jan Plesnik. Constrained weighted matchings and edge coverings in graphs. *Discrete Applied Mathematics*, 92(2-3):229–241, Jun. 1999.
- [25] CBC. <http://www.coin-or.org/projects/Cbc.xml>.
- [26] OpenSPARC T1. <http://www.oracle.com/technetwork/systems/opensparc/index.html>.
- [27] Synopsys Design Compiler. <http://www.synopsys.com>.
- [28] NanGate FreePDK45 Generic Open Cell Library. <http://www.si2.org/openeda.si2.org/projects/nangatelib>, 2008.
- [29] Cadence SOC Encounter. <http://www.cadence.com>.