DSAR: DSA aware Routing with Simultaneous DSA Guiding Pattern and Double Patterning Assignment *

Jiaojiao Ou[†], Bei Yu[‡], Xiaoqing Xu[†], Joydeep Mitra[♯], Yibo Lin[†], David Z. Pan[†]
[†]ECE Department, The University of Texas at Austin
[‡]CSE Department, The Chinese University of Hong Kong
[♯]Mentor Graphics Corporation

ABSTRACT

Directed self-assembly (DSA) is a promising solution for fabrication of contacts and vias for advanced technology nodes. In this paper, we study a DSA aware detailed routing problem, where DSA guiding pattern assignment and guiding pattern double patterning (DP) compliance are resolved simultaneously. We propose a net planning technique, which pre-routes some nets based on their bounding box positions, to improve both metal layer and via layer qualities. We also introduce a new routing graph model with DSA and DP design rule considerations. The DSA and DP aware detailed routing is then performed based on the net planning result, followed by a post-routing optimization on DSA guiding pattern assignment and decomposition. The experimental result demonstrates that our proposed approach can achieve promising DSA and DP friendly layout, i.e., conflict free on DSA guiding pattern with double patterning assignment for via layer. In addition, our proposed detailed router is able to effectively reduce 20% via number and 15% total wirelength than one recent DSA aware detailed router.

1. INTRODUCTION

With the delay of extreme ultraviolet lithography (EUV), industry is heavily looking for other lithography alternatives, such as multiple patterning lithography (MPL) for $7nm$ technology node and beyond [16, 25, 33]. Besides, they are also looking for other options such as multiple e-beam or directed self-assembly (DSA) lithography, to serve as the next generation lithography techniques to extend $193nm$ immersion (193i) lithography further or combine with EUV to reduce manufacturing cost [12, 21]. Recently, one dimensional (1D) design has been used to simplify the design rule explosion in advanced technology nodes. Litho-friendly 1D standard cell design method, place and route algorithm have been developed to improve chip yield and performance [2, 10, 21].

With the scaling of technology nodes, the via density has increased dramatically on routing layers, leading to triple and even quadruple patterning for via layer printing with conventional 193i lithography [13, 21]. The situation becomes even worse for the 1D design where more vias may be introduced due to extreme regular routing patterns. As a result, the cost grows tremendously for the mask manufacturing. On the other hand, the number of masks for contacts/vias layer can be reduced by DSA because of its pitch multiply ability to group several contacts/vias in a single DSA guiding pattern [5, 6, 13, 15, 19, 20, 26, 28, 29, 32]. For example, in order to print the via layer in Figure 1(a), at least four masks with 193i lithography should be used, as shown in Figure 1(b). If DSA is applied on via layer, since the dense distributed vias can be grouped in the same DSA pattern, then only two masks are required for printing of DSA guiding patterns (GP), as shown in Figure 1(c). In this way, both the cost and number of masks can be greatly reduced.

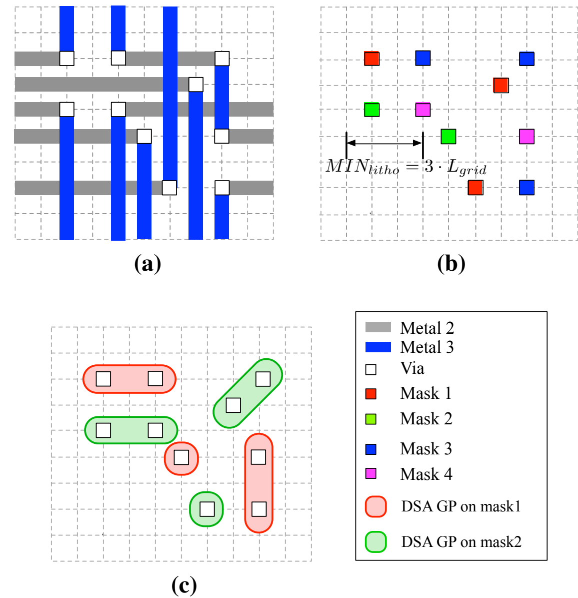


Figure 1: (a) Original layout; (b) Print via layer with quadruple patterning; (c) Print via layer with DSA and double patterning.

As an emerging lithography technique, DSA can formulate ultra regular cylinders inside the guiding patterns. The pitch of cylinders is multiplied when several cylinders are formulated inside the same guiding pattern. The self-assemble process is enabled by micro-phase separation of Block Copolymer (BCP) material, and it is directed by different guiding pattern shapes to generate different contacts/vias patterns. DSA is considered as a potential complementary hole shrinking technique to EUV lithography [5, 19]. Nevertheless, DSA also suffers from placement error caused by the

*This work is supported in part by NSF, SRC, and CUHK Direct Grant for Research.

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'17, March 19-22 2017, Portland, OR, USA
© 2017 ACM. ISBN 978-1-4503-4696-2/17/03. . . $15.00
DOI: http://dx.doi.org/10.1145/3036669.3036677

mismatch between final and target patterns. For instance, it is discovered that the placement error may be quite large for complex DSA group patterns [19], thus some regular DSA group patterns are preferred due to yield consideration. However, due to the limited patterns of DSA, it is difficult to cover the randomly distributed contacts and vias in conventional design by a single guiding pattern mask. Therefore, multiple patterning for DSA guiding patterns is a straightforward and natural extension.

There are several works on DSA guiding pattern assignment and decomposition problem. [17, 22, 23, 30] studied the DSA guiding pattern assignment and redistribution of cut masks in 1D design; [20] first addressed the challenges of applying DSA to contacts and vias, and proposed the initial ideas of multiple patterning in DSA; [3, 4, 14, 31] investigated the DSA guiding pattern assignment, and proposed different methods to decompose these patterns in order to reduce variations. It should be noted that decomposition conflicts may not be completely removed if vias in the design is fixed, so considering DSA patterns during design stage becomes necessary. [7] proposed an SAT algorithm to optimize the contact topology of 1D standard cell library in order to reduce complexity of DSA guiding patterns. [8] proposed a detailed routing method to assign DSA guiding patterns to vias during the detailed routing stage. [9] considered the DSA pattern during the redundant via insertion stage, and solved this problem by using integer linear program (ILP) and maximum independent set method; [24] extended the DSA aware redundant via insertion problem, and solved the guiding pattern assignment and redundant via insertion simultaneously. [27] proposed a method to remove the grouping conflicts between the standard cell boundaries in the post-placement stage.

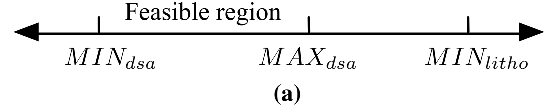
However, none of the previous research consider DSA and multiple patterning constraints simultaneously during detailed routing stage. In this paper, we investigate detailed routing algorithms considering DSA and double patterning (DSA+DP) on guiding patterns. The purpose of this study is to obtain a detailed routing layout with DSA and double patterning friendly via distribution, while the impact on conventional metrics such as wirelength and via number are minimized as well. Our major contributions of this paper can be summarized as follows:

- This is the first work considering DSA with double patterning for DSA guiding patterns on via layer in detailed routing stage to the best of our knowledge.
- A pre-route net planning algorithm is proposed to improve routing quality in terms of total via number and wirelength.
- A routing graph model with DSA+DP considerations is developed for detailed routing.
- The DSA guiding pattern assignment and decomposition for via layer can be performed simultaneously in the routing stage.
- A post-route DSA guiding pattern assignment and decomposition method is adopted to further improve the result.

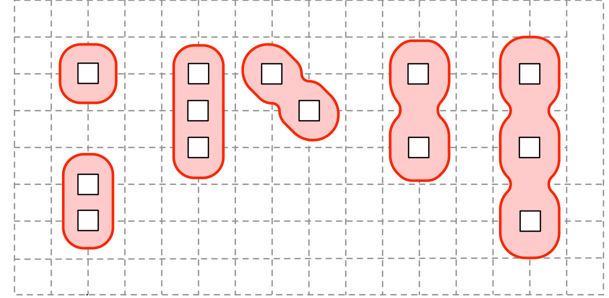
The rest of the paper is organized as follows: Section 2 introduces the DSA pattern constraints and problem formulation. Section 3 presents the details of our proposed method. Section 4 demonstrates the experimental results. Then Section 5 concludes the paper.

2. PRELIMINARIES

In this section, we introduce the DSA related design rules and present our problem formulation. The design is assumed as gridded design.



(a)



(b)

Figure 2: (a) DSA design rules: Vias whose distance is within $[MIN_{dsa}\ ,\ MAX_{dsa}]$ can be grouped in the same guiding template; otherwise they can not be grouped. (b) Feasible regular DSA patterns. As the last three patterns have larger pitch than the natural pitch of DSA material, their guiding templates will have complex shapes in order to obtain higher confinement on the final patterns. Without loss of generality, all the guiding templates are illustrated as rectangles in the following sections.

2.1 DSA Design Rules

The minimum and maximum pitches that DSA material can obtain are defined as MIN_{dsa} and MAX_{dsa}. The MIN_{dsa} is almost the same to the natural pitch L_O of DSA material. Only when two vias are within the region between MIN_{dsa} and MAX_{dsa}, can they be grouped in the same DSA group. Otherwise, they have to be decomposed into different masks when the distance is smaller than the conventional minimum lithography distance MIN_{litho}. In this paper, in order not to lose any generality, we assume the size of the grids is L_{grid}, and the minimum lithography distance $MIN_{litho} \geq 3 \times L_{grid}$. And $MIN_{dsa} \leq L_{grid}$, $2 \times L_{grid} \leq MAX_{dsa} \leq 3 \times L_{grid}$. The above assumptions are illustrated in Figure 2(a). Moreover, as the non-collinear and non-manhattan aligned DSA patterns are not reliable with 193i lithography, and fork or cycle shaped patterns are difficult to synthesis, thus these kind of patterns are forbidden. In addition, the number of contacts/vias in a single DSA guiding pattern are also restricted. Therefore, the regular and simple DSA patterns are adopted in this work to improve the robustness and yield of DSA lithography, as shown in Figure 2(b). When the distance between adjacent holes is different from the natural pitch of DSA material, more confinement is required on the guiding template. Thus, the shapes of masks for the last three guiding templates are more complex than the first three.

2.2 Forbidden Via Distribution

For unidirectional design, cut masks are often used to cut off the metal lines to formulate logic connections. But it should be noted that there are minimum area constraints for the cut masks. As illustrated in Figure 3(a), the tip-to-tip design with two adjacent vias on different nets might violate the design rules. Therefore, this kind of adjacent via distributions on different nets are forbidden. However, the minimum area violation can be avoided for adjacent vias on the same nets, as shown in Figure 3(b).

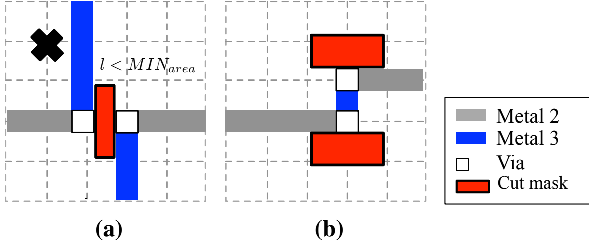


Figure 3: (a) The cut mask violates the minimum area design rule for adjacent vias on different nets. (b) If the adjacent vias are on the same net, the cut mask will not violate the rule.

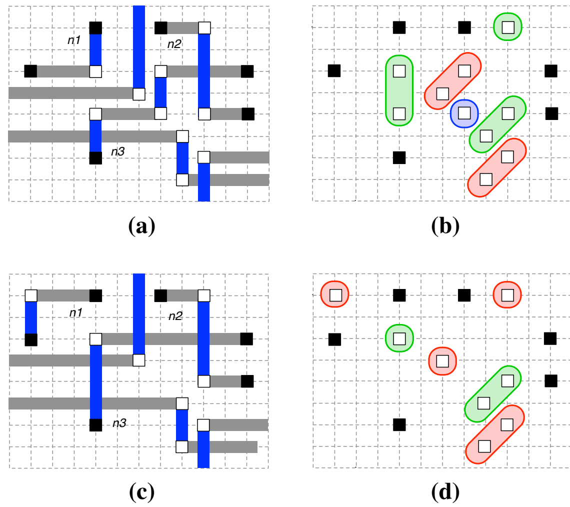


Figure 4: (a) Initial detailed routing without DSA consideration. (b) Three masks are required for via layer. (c) Redistribute the via by rerouting net 1 and net 3 to make the via layer DSA friendly. (d) Only two masks are used with DSA consideration.

2.3 Problem Formulation

Figure 4 gives an example of DSA incompatible routing and DSA friendly routing. Without DSA consideration on the via layer, the via distribution in Figure 4(a) requires three DSA masks to print the vias as in Figure 4(b). By rerouting net 1 and net 3 with DSA consideration, we can pattern the vias with only two DSA masks, as illustrated in Figure 4(d). The problem formulation of our DSA and double patterning aware routing (DSAR) is as follows:

Problem 1 (DSAR). *Given a placed netlist with source/target pins, a set of feasible DSA patterns and corresponding design rules, we will perform detailed routing and DSA+DP guiding pattern assignment simultaneously, and minimize the number of unroutable nets and total wire length at the same time.*

3. DSAR: DSA+DP AWARE ROUTING

This section introduces a pre-route net planning algorithm and a new routing graph model. It also provides the details of algorithms that have been used.

3.1 Pre-route Net Planning

Net planning is one of the important problems in the detailed routing stage, which has been studied in many different ways. In the following, we present a new algorithm to estimate the via distribution with DSA+DP considerations and net routability before performing the A* path search in detailed routing. The proposed algorithm generates a pre-route net path assignment based on the bounding box locations of all the nets. By determining the routing path for nets that do not have too much interference or impact on other nets with DSA+DP constraints in advance, the pre-route net planning could be able to reduce the via number, the total wire-length and conflict number.

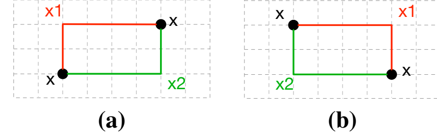


Figure 5: (a)(b) Corners for net x.

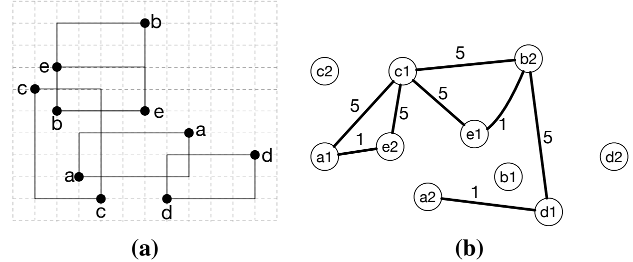


Figure 6: (a) Bounding boxes of nets. (b) Conflict graph for the two corners of all nets.

3.1.1 *Conflict Graph Construction for Nets*

We first construct a conflict graph for the possible vias based on the bounding box locations of each net. As for unidirectional routing, different metal layers have certain routing direction, running either horizontally or vertically. Therefore, it is reasonable to take the jogs of the bounding box as a via between different metal layers. So besides the source and target pins, each bounding box has two corners, denoted as x_1 and x_2. The first one indicates the upper corner, and the latter one indicates the lower corner, as shown in Figure 5. The two corners are used as the vertices in the conflict graph. One corner may contain one to three vias depending on the pin locations. To make it simple, we assume that no via is inserted on the pins in the example. A conflict edge will be added between vertices when their distance is less than the minimum lithography distance. For edge whose vertices can not be grouped in the same DSA guiding pattern, a larger weight is assigned to it. Similarly, a smaller weight is assigned to edge whose vertices can be grouped in the same DSA guiding pattern.

Figure 6 shows an example of conflict graph construction, where there are fives nets and ten vertices for each corner of all the nets. As the two possible vias in the lower corner a_2 of net a and upper corner d_1 of net d can be grouped in the same DSA pattern, a smaller weight is assigned to the edge between a_2 and d_1. The above rules can be applied to other vertices and edges in the same way.

3.1.2 *Conflict Graph Bipartization*

In our net planning algorithm, we want to determine the routing path for as many nets as possible. With the conflict graph, the target is to find the maximized number of vertices which are 2-colorable, by grouping vias in the same DSA group. Therefore, this problem

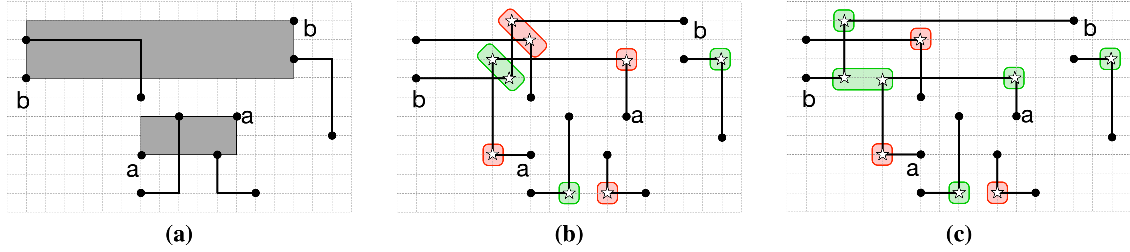


Figure 8: (a) Bipartization result, net a and b are undetermined. (b) Route b first results in longer wire length. (c) Route a first results less wire length.

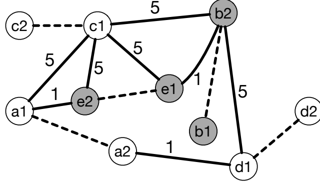


Figure 7: Conflict graph constraints.

Table 1: Pre-route Net Planning Notations

c_v	vertices deletion coverage for vertex v
s_v	color assignment for vertex v
α_v	cost of vertex v
N	total number of vertices
E_s	solid edge set
E_d	dashed edge set
V_f	forbidden vertex set

can be formulated as a constrained DSA+DP vertex bipartization algorithm, which converts the conflict graph to a bipartite graph by deleting minimized number of vertices.

It should be noted that at most one corner can be selected for each net. A net corner is strictly forbidden if it goes through the source/-target pins of another net. Other constraints for conflict graph are demonstrated in Figure 7. The dashed line means that the two corners belong to the same net, and at most one corner can exist at the same time. The dashed lines will not be counted as the conflict edge during the graph bipartization. The shadowed vertices indicate that the corresponding corner goes through the source/target pins of another net, so these vertices have to be deleted in advance. The ideal case is that there exists a set of valid corners for each net that are 2-colorable for the original conflict graph. ILP formula (1) is proposed to solve the problem. Some notations are explained in Table 1.

$$\textbf{min} \quad \sum_{i=1}^{N} (\alpha_i \times c_i) \tag{1a}$$

$$\begin{aligned}
\textbf{s.t.} \quad & s_v + s_u + (c_v + c_u) \geq 1, && \forall \{v,u\} \in E_s, && \text{(1b)}\\
& s_v + s_u - (c_v + c_u) \leq 1, && \forall \{v,u\} \in E_s, && \text{(1c)}\\
& c_v + c_u \geq 1, && \forall \{v,u\} \in E_d, && \text{(1d)}\\
& c_v = 1, && \forall v \in V_f. && \text{(1e)}
\end{aligned}$$

c_v is a binary variable indicating the vertices deletion coverage for v. $c_v = 1$ means vertex v is deleted. The cost of c_v is denoted as α_v, which is calculated as $\alpha_v = \frac{1}{\sum_{e \in V} Cost_e}$. It is calculated as the reciprocal of the sum of connected edge cost. With the cost,we prefer to delete the most congested and DSA unfriendly vertices to avoid decomposition conflicts during the bipartization process. s_v is also a binary variable indicating the color assignment for vertex v. $s_v = 0$ means that v is assigned with color 0, and vice versa for $s_v = 1$. And N is the total number of vertices. The objective is to delete as few vertices as possible with minimized cost. The first two constraints forbid both endpoints of a conflict edge to have the same color while none of them is deleted. The third constraint forbids the two corners of the same net to exist at the same time. The forth constraint forbids the vertices if corresponding corner of its bounding box occupies the source/target pin locations of other nets.

3.1.3 DSA+DP Net Ordering

Due to the congested bounding box distribution, the routing path of some nets can not be decided with ILP. Therefore, these nets will be routed in the detailed routing stage. Before that, a net ordering method is proposed based on previous ILP result. Generally, the net order of the undetermined nets are determined in the ascending order based on the size of bounding box of each net. However, if the bounding box size is the same, the net with more overlaps with other nets will be routed first. Figure 8 shows that routing paths of nets a and b are not determined, net a is routed first since it has smaller bounding box.

3.2 Detailed Routing

3.2.1 Routing graph model

Since we only consider the DSA guiding pattern decomposition on via layer where the metal layers are all unidirectional, previous routing graph model for multiple patternings [18] may no longer be suitable for the DSA+DP aware detailed routing. To avoid the generations of DSA guiding pattern conflicts, we propose the following DSA+DP aware routing graph model to perform DSA guiding pattern assignment and decomposition simultaneously.

Without loss of any generality, we assume that the vertical wire indicates metal 2, and the horizontal wire indicates metal 3. A via is inserted when the routing direction changes. A routing box for each routing grid is used to indicate the inserted via and its color assignment, as illustrated in Figure 9(a). There are four outlets on the sides of the routing box to denote the routing direction of the net. Inside the routing box, several lines are used to indicate the routing paths to the next grid. The horizontal and vertical lines are used for routing metals on different layers.The diagonal lines indicate the direction change of the routing path, and the direction change means a via is inserted between different metal layers. There are

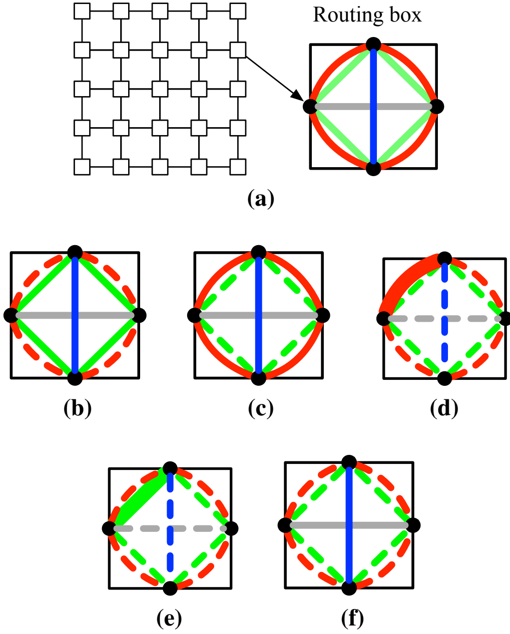


Figure 9: (a) Routing graph model. The inner path inside the routing box will be marked as dashed lines when it is forbidden. (b) Red via is forbidden. (c) Green via is forbidden. (d) Red via is inserted. (e) Green via is inserted. (f) All vias are forbidden.

four routing direction changes in each routing box. In order to show the DSA guiding pattern assignment to the inserted via, and the decomposition of these templates, we use color green and red to indicate each direction change. If the line is dashed, then this color is not allowed in this routing box. Figures 9(b)–9(f) demonstrate different routing box states, where Figures 9(b) and 9(c) forbid a certain via color assignment inside the routing box, Figures 9(d) and 9(e) show that if a via is inserted, the metal lines and any other vias are forbidden in the routing box. Figure 9(f) shows that vias are forbidden but metal lines are allowed in the routing box.

With the definition of the routing box, the cost from current grid to next grid (c_i^{next}) can be calculated based on the relative locations of both grids.

$$c_i^{next} = \begin{cases} c_{via} + L_{grid}, & d_i \neq d_{next}, \\ c_m + L_{grid}, & d_i = d_{next}. \end{cases} \tag{2}$$

Besides the basic wire length cost L_{grid}, we have the metal cost c_m and via cost c_{via} to be added to the cost based on the direction of current grid d_i and next grid d_{next}. A larger weight M should be assigned to the cost if the routing direction is forbidden.

Once a net is routed, we need to update the status for corresponding routing boxes. It is observed that once a via has been inserted in the routing box, all the other routing paths should be forbidden. And if a vertical metal line goes through a routing box, the horizontal metal line is still valid to go through the routing box, but all via paths are forbidden. As shown in Figure 10(a), when a green via is inserted, the neighboring routing boxes of the inserted via are marked as via-forbidden. For other routing boxes whose distance is less than the minimum lithography distance, but they can not be grouped with current via in the same DSA pattern, then these routing boxes can only be assigned to the opposite colors. If the grid distance to the inserted via is larger than minimum lithography distance, no coloring constraints is required. Figure 10(b) gives an example for the updated routing boxes. When both via colors are

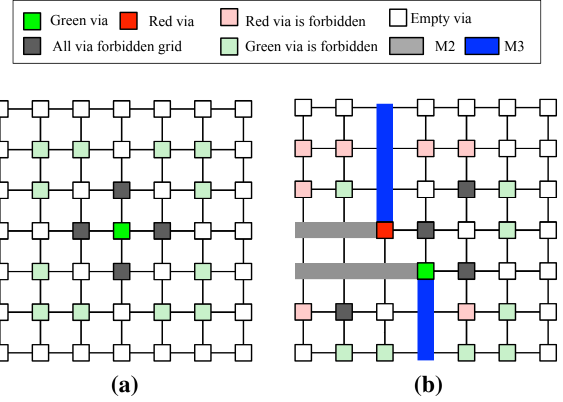


Figure 10: (a) Nearby routing box state update. (b) Example layout shows the updated routing box state.

forbidden in a routing box, then it is marked as infeasible routing box for all vias.

3.2.2 General Cost Function

Assume current grid is i, and next grid is $next$. Let $cost_s^i$ denote the cost from source to i. Let $dist_{next}^t$ denote the estimated distance from $next$ to target. Let $usage(i)$ denote the number of nets in the grid. Besides, we also need to consider the grid conflicts cost and related history cost for the A* search. Thus, let $h(i)$ be the current history cost of grid i, $h(i)'$ be the previous history cost of grid i. And $h_{dsa}(i)$ is the DSA violation history.

$$h(i) = h(i)' + A \times usage(i) + B \times h_{dsa}(i). \tag{3}$$

Therefore, according to the routing box cost explained in Section 3.2.1, the current grid cost $c(i)$ and estimated net cost $p_i(s,t)$ so far can be calculated as follows:

$$c(i) = cost_s^i + c_i^{next} + h(i), \tag{4}$$

$$p_i(s,t) = c(i) + \sigma \times dist_{next}^t. \tag{5}$$

We assign a higher value to σ if the HPWL of current net is less than the estimated cost to help the A* search.

$$\sigma = \begin{cases} 1, & c(i) \leq l_{HPWL}, \\ 1 + \dfrac{c(i)}{HPWL}, & c(i) > l_{HPWL}. \end{cases} \tag{6}$$

For A* search, we need to search the neighbors of current grid, and create a priority queue to store the neighbors with estimated net cost. If the nearby grid does not exist in the priority queue, or the new estimated path cost is smaller than the previous one, we update the priority with the new estimated path cost. The search continues until we reach the target pin of current net.

3.2.3 DSA+DP aware Detailed Routing

The detailed routing algorithm is illustrated in Algorithm 1. A* search scheme is adopted in this work. We update the corresponding grid cost after a net is routed, and use a queue to store all the violated grids and all the nets information in these grids. If a net path generates a violation, the net path will be rip-up, and its nearby grid status should be updated. The net will be rerouted to remove the violation. If a new violation is generated after the rip-up and reroute, the new conflict should be pushed into the queue. This rip-up and reroute iteration continues until the violation queue is empty or the runtime exceeds the maximum time requirement.

Algorithm 1 DSA+DP aware detailed routing

Input: Netlists from net planning algorithm.
Output: Routed nets with DSA friendly via layer.
1: Route determined nets;
2: Update grids cost;
3: Initial routing iteration;
4: Q ← nets in violated grids;
5: **while** !Q.empty() **do**
6: $g(i) \leftarrow$ Q.pop(); $Nets \in g(i)$;
7: **for** each net $k \in Nets$ **do**
8: Pre-route with cost evaluation;
9: **end for**
10: Rip-up net k that has maximum cost improvement;
11: Route net k;
12: **for** each grid $g(j)$ of net k **do**
13: Update grid cost;
14: **if** $g(j)$ is violated **then**
15: Q ← nets in this $g(j)$;
16: **end if**
17: **end for**
18: **if** $g(i)$ is still violated **then**
19: Q ← $g(i)$;
20: **end if**
21: **end while**

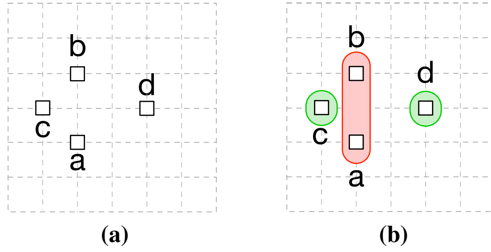


Figure 11: (a) Example layout for post routing optimization. (b) Optimal guiding pattern assignment and decomposition for the example layout.

3.3 Post Routing Optimization

In the post-routing stage, we propose a method to further minimize the DSA pattern cost and conflicts by re-assigning DSA guiding patterns to the via layer. For example, the original via pattern is shown in Figure 11(a), the optimized DSA guiding pattern assignment and decomposition is shown Figure 11(b). To find the optimized solution, we will first construct a new conflict graph for the via layer, as illustrated in Figure 12(a). The vertices represent the vias. An edge is added between any two vias when their distance is less than the minimum lithography distance. In addition, the edge is marked with black color if its vias can be grouped in the same DSA group, and the edge is marked with red color if its vias can not be grouped in the same DSA group. Because larger DSA groups can increase mask complexity and placement error between the target via and final via, thus minimizing large DSA groups and the number of conflicts at the same time are the primary goals for the post-routing optimization.

This problem can be formulated as the edge bipartization problem: convert the conflict graph to a bipartite graph by deleting minimized numbers of edges. This formulation is slightly different from the pre-route net planning formulation, as shown in Formula (7).

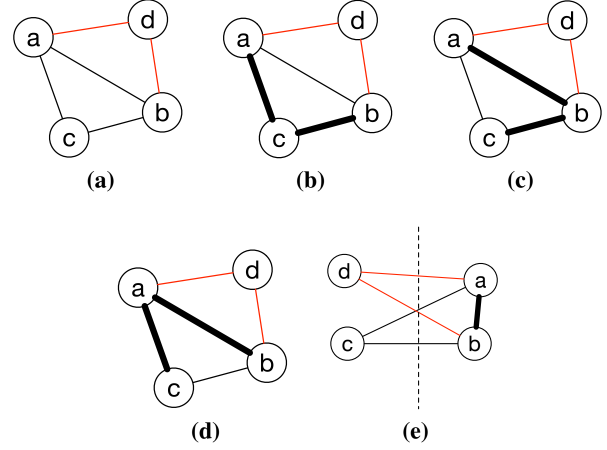


Figure 12: (a) Conflict graph for post routing optimization of the example layout. Because vias a, b and c cannot be grouped in the same DSA guiding template, therefore: (b) Edge ac and bc can not be deleted at the same time. (c) Edge ab and bc can not be deleted at the same time. (d) Edge ab and ac can not be deleted at the same time. (e) The conflict graph can be converted to a bipartite graph by deleting edge ab with minimum cost.

$$\textbf{min} \quad \sum e_{v_iv_j} + M \cdot \sum e_{v_kv_h} \tag{7a}$$

$$\textbf{s.t.} \quad t_{v_i} + t_{v_j} + e_{v_iv_j} \geq 1, \qquad \forall \{e_{v_iv_j}\} \in E, \tag{7b}$$

$$t_{v_i} + t_{v_j} - e_{v_iv_j} \leq 1, \qquad \forall \{e_{v_iv_j}\} \in E, \tag{7c}$$

$$e_{v_iv_j} + e_{v_iv_k} \leq 1, \qquad \forall (i,j,k) \text{ infeasible}, \tag{7d}$$

$$e_{v_iv_j} + e_{v_jv_k} + e_{v_kv_h} \leq 2, \qquad \forall (i,j,k),(j,k,h) \text{ groupable}, \tag{7e}$$

$$e_{v_iv_j}, t_{v_i} \in \{0,1\}. \tag{7f}$$

The first item in the objective function is the sum of edges with groupable vias, while the second item is the sum of edges with un-groupable vias. v_i represents via i, while $e_{v_iv_j}$ is a binary variable to indicate the edge between v_i and v_j. $e_{v_iv_j} = 1$ means that the edge is deleted. But if v_i and v_j can not be grouped in the same DSA pattern, a conflict is generated. Therefore, in order to discourage deleting the un-groupable edge, a large coefficient M is assigned to these edges in the objective function. t_{v_i} indicates which masks that v_i has been assigned to. The first two constraints will try to divide the vias on the same edge into two masks. The third constraint will avoid the forbidden DSA patterns, such as forks and circles. And the forth constraint restricts the sizes of feasible DSA patterns. As illustrated in Figures 12(b)–12(c), the bold edges can not be deleted at the same time to avoid infeasible DSA patterns. And the optimal solution can be achieved by deleting the edge between a and b to make the conflict graph a bipartite graph, as shown in Figure 12(e).

3.4 Overall Flow

Figure 13 shows the overall flow of our DSA+DP aware detailed routing algorithm. First, the algorithm input consists of blockages, net list with source and target pins, a set of DSA pattern constraints and corresponding design rules. Next, we perform the pre-route net planning to determine the routing paths for some nets. For nets whose routing paths can not be decided during the pre-route stage,

Table 2: Benchmark Statistics

bench	#net	#pin	Grid size
ecc	1671	3342	436×446
efc	2219	4438	406×421
ctl	2706	5412	496×503
alu	3108	6216	406×408
div	5813	11626	636×646
top	22201	44402	1176×1179

the rip-up and reroute based A* search algorithm is performed on our DSA+DP aware routing graph model to find the routing paths for them. After that, the post routing optimization algorithm is applied on the via layer to further reduce template cost.

4. EXPERIMENTAL RESULTS

We implemented the proposed algorithms in C++. All the experiments are performed on a 3.4 GHz Intel workstation with 32GB memory. The state of art optimization tool GUROBI 6.5 [11] is used as the ILP solver. The benchmarks are the synthesized OpenSparc T1 design [1]. It is assumed that only Metal 2 (M2) and Metal 3 (M3) layers are available for routing, where M2 and M3 run horizontally and vertically. The benchmarks are grid-based, and statistics of the benchmarks are shown in Table 2.

In order to demonstrate the effectiveness of our approach, we implemented the conventional 1D detailed router without DSA considerations, and also the DSA-aware detailed router proposed in [8], in which DP is not considered. Although the DSA patterns and design rules are different in [8], we adjusted the method with our own design rules and feasible DSA patterns. The parameters A and B in Equation (3) are set as 1 and 10 respectively.

Table 3 compares experimental results of conventional 1D detailed router, DSA-aware detailed router in [8], and our proposed DP+DSA aware detailed router (DSAR). In the table, “#Via” is the total via number; “WL” is the total wire length; “#CFLT” is number of the DP+DSA conflicts; while “CPU” is the runtime in seconds.

As shown in the table, our DSAR can effectively reduce via number by around 1% and 20% compared to conventional 1D detailed router and DSA aware detailed router without double patterning considerations [8]. In addition, DSAR outperforms [8] in terms of total wire length “WL” by reducing it by 15%. The wirelength of DSAR increases less than 1% compared to conventional 1D routing. These demonstrate that our proposed detailed routing algorithm is quite effective to reduce the wire length and via number than DSA aware detailed router without DP consideration, and with little impact compared to conventional 1D router. Moreover, compared to conventional 1D router, our DSAR can reduce 269 more DP+DSA conflicts between DSA guiding patterns, and reduce 8 conflicts than DSA router without double patterning considerations, which demonstrates the effectiveness of our proposed algorithms on DSA+DP considerations. The runtime is a little longer because of more iterations to resolve double patterning conflicts and more search spaces during A* search process. Besides, as all the algorithms can achieve 100% routability, thus routability is not shown in the comparison table.

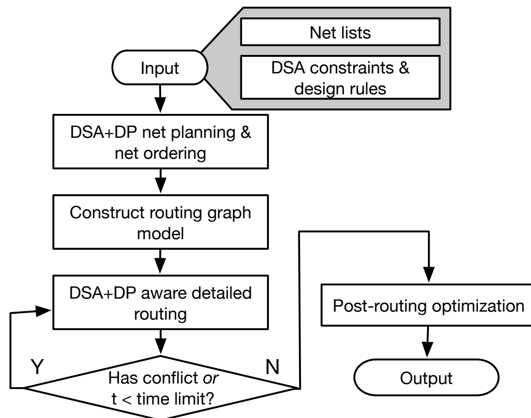


Figure 13: Overall flow of detailed routing with DSA+DP consideration.

The impact of our proposed net planning method is also analyzed in terms of total via number, total wirelength and runtime. The comparisons are illustrated in Figure 14. We can observe from Figures 14(a) and 14(b) that, with our proposed pre-route net planning, the via number and the wire length can be effectively reduced by 19% and 8%, respectively. Note that with more pre-routed nets the A* search based maze routing takes more time to search for the neighboring grids, thus the runtime has been slightly increased by around 7%, as shown in Figure 14(c). It indicates that with preroute net planning, the solution quality can be improved at a cost of small runtime increase. Overall, we can still claim that the proposed net planning method is effective to reduce the total number of inserted vias and wirelength.

5. CONCLUSION

With the pitch multiplication ability, DSA can be applied on the printing of contacts/vias layer to reduce the number of masks. In this paper, we have proposed an effective DSA aware detailed routing algorithm with double patterning consideration on guiding patterns. We have also proposed a pre-route net planning algorithm to improve the routing quality in terms of via number and total wirelength. The detailed routing was performed on a new routing model with DSA+DP consideration, followed by a post routing optimization algorithm to further improve the DSA guiding pattern assignment and decomposition. The experimental results show that our proposed algorithm effectively reduces coloring conflicts on via layer with DSA and double patterning considerations.

6. REFERENCES

[1] OpenSPARC T1. http://www.oracle.com/technetwork/systems/opensparc/index.html.

[2] V. Axelrad, K. Mikami, M. Smayling, K. Tsujita, and H. Yaegashi. Characterization of 1D layout technology at advanced nodes and low k1. In *Proceedings of SPIE*, volume 905213–905213, 2014.

[3] Y. Badr, A. Torres, and P. Gupta. Mask assignment and synthesis of DSA-MP hybrid lithography for sub-7nm contacts/vias. In *ACM/IEEE Design Automation Conference (DAC)*, pages 70:1–70:6, 2015.

[4] Y. Badr, J. A. Torres, Y. Ma, J. Mitra, and P. Gupta. Incorporating DSA in multipatterning semiconductor manufacturing technologies. In *Proceedings of SPIE*, volume 9427, 2015.

[5] I. Bita, J. K. W. Yang, Y. S. Jung, C. A. Ross, E. L. Thomas, and K. K. Berggren. Graphoepitaxy of self-assembled block copolymers on two-dimensional periodic patterned templates. *Science*, 321(5891):939–943, 2008.

[6] L.-W. Chang, X. Bao, B. Chris, and H.-S. P. Wong. Experimental demonstration of aperiodic patterns of directed self-assembly by block copolymer lithography for random logic circuit layout. In *IEEE International Electron Devices Meeting (IEDM)*, pages 33.2.1–33.2.4, 2010.

[7] Y. Du, D. Guo, M. D. F. Wong, H. Yi, H.-S. P. Wong, H. Zhang, and Q. Ma. Block copolymer directed self-assembly (DSA) aware contact layer optimization for 10 nm 1D standard cell library. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 186–193, 2013.

[8] Y. Du, Z. Xiao, M. D. F. Wong, H. Yi, and H.-S. P. Wong. DSA-aware detailed routing for via layer optimization. In *Proceedings of SPIE*, volume 9049, 2014.

[9] S.-Y. Fang, Y.-X. Hong, and Y.-Z. Lu. Simultaneous guiding template optimization and redundant via insertion for directed self-assembly. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 410–417, 2015.

Table 3: Routing Result Comparison

bench	Conventional 1D router				SPIE-14 [8]				DSAR			
	#Via	WL	#CFLT	CPU(s)	#Via	WL	#CFLT	CPU(s)	#Via	WL	#CFLT	CPU(s)
ecc	3243	37452	1	4.26	4068	44193	0	10.56	3269	37801	0	13.66
efc	5059	48712	2	19.9	6162	57024	1	27.5	5102	49686	0	26.69
ctl	5782	60059	1	22.02	6989	69397	0	57.9	5782	60574	0	56.94
alu	7303	65139	9	20.19	8999	77322	2	56.12	7489	67140	0	45.87
div	12897	124108	11	97.03	15689	144316	1	246.09	12974	125734	0	278.47
top	49347	424869	245	1558.32	57744	491182	4	4382.50	48044	423194	0	4133.30
Sum.	83631	760339	269	1721.72	99651	883434	8	4780.67	82660	764129	0	4554.93
Norm.	1.01	0.99	–	0.38	1.20	1.15	–	1.05	1.00	1.00	–	1.00

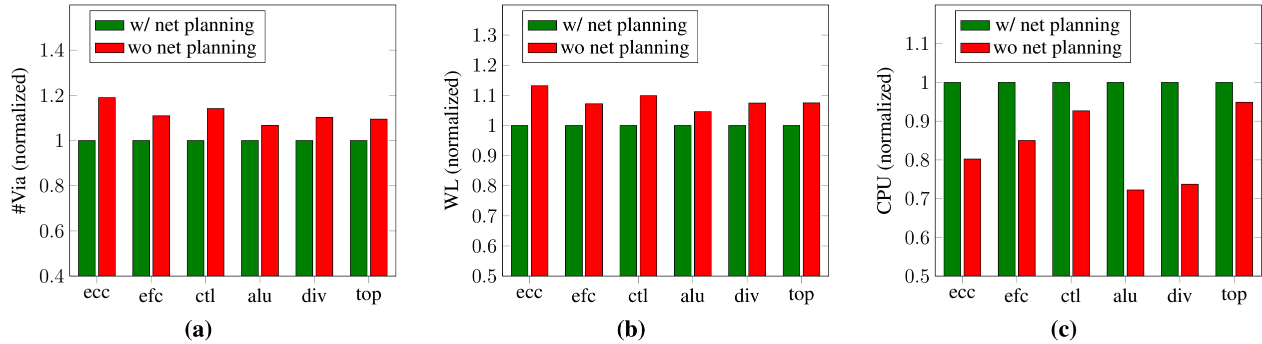


Figure 14: Results comparison between algorithm with net planning and without net planning for (a) total via number, (b) total wirelength, (c) runtime. The data has been normalized for comparison.

[10] W. Gillijns, S. Sherazi, D. Trivkovic, B. Chava, B. Vandewalle, V. Gerousis, P. Raghavan, J. Ryckaert, K. Mercha, D. Verkest, et al. Impact of a SADP flow on the design and process for N10/N7 metal layers. In *Proceedings of SPIE*, volume 9427, 2015.

[11] Gurobi Optimization Inc. Gurobi optimizer reference manual. http://www.gurobi.com, 2014.

[12] R. Ikeno, T. Maruyama, S. Komatsu, T. Iizuka, M. Ikeda, and K. Asada. A structured routing architecture and its design methodology suitable for high-throughput electron beam direct writing with character projection. In *ACM International Symposium on Physical Design (ISPD)*, pages 69–76, 2013.

[13] I. Karageorgos, J. Ryckaert, M. C. Tung, H. S. P. Wong, R. Gronheid, J. Bekaert, E. Karageorgos, K. Croes, G. Vandenberghe, M. Stucchi, and W. Dehaene. Design strategy for integrating DSA via patterning in sub-7nm interconnects. In *Proceedings of SPIE*, volume 9781, 2016.

[14] J. Kuang, J. Ye, and E. F. Y. Young. Simultaneous template optimization and mask assignment for DSA with multiple patterning. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 75–82, 2016.

[15] A. Latypov, T. H. Coskun, G. Garner, M. Preil, G. Schmid, J. Xu, and Y. Zou. Simulations of spatial DSA morphology, DSA-aware assist features and block copolymer-homopolymer blends. In *Proceedings of SPIE*, volume 9049, 2014.

[16] L. Liebmann, A. Chu, and P. Gutwin. The daunting complexity of scaling to 7nm without EUV: Pushing DTCO to the extreme. In *Proceedings of SPIE*, volume 9427, 2015.

[17] Z.-W. Lin and Y.-W. Chang. Cut redistribution with directed self-assembly templates for advanced 1-D gridded layouts. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 89–94, 2016.

[18] Q. Ma, H. Zhang, and M. D. F. Wong. Triple patterning aware routing and its comparison with double patterning aware routing in 14nm technology. In *ACM/IEEE Design Automation Conference (DAC)*, pages 591–596, 2012.

[19] Y. Ma, J. Lei, J. A. Torres, L. Hong, J. Word, G. Fenger, A. Tritchkov, G. Lippincott, R. Gupta, N. Lafferty, Y. He, J. Bekaert, and G. Vanderberghe. Directed self-assembly (DSA) grapho-epitaxy template generation with immersion lithography. In *Proceedings of SPIE*, volume 9423, 2015.

[20] Y. Ma, J. A. Torres, G. Fenger, Y. Granik, J. Ryckaert, G. Vanderberghe, J. Bekaert, and J. Word. Challenges and opportunities in applying grapho-epitaxy DSA lithography to metal cut and contact/via applications. In *Proceedings of SPIE*, volume 9231, 2014.

[21] A. Mallik, J. Ryckaert, A. Mercha, D. Verkest, K. Ronse, and A. Thean. Maintaining Moore's law - enabling cost-friendly dimensional scaling. In *Proceedings of SPIE*, volume 9422, 2015.

[22] J. Ou, B. Yu, J.-R. Gao, and D. Z. Pan. Directed self-assembly cut mask assignment for unidirectional design. *Journal of Micro/Nanolithography, MEMS, and MOEMS (JM3)*, 14(3), 2015.

[23] J. Ou, B. Yu, J.-R. Gao, D. Z. Pan, M. Preil, and A. Latypov. Directed self-assembly based cut mask optimization for unidirectional design. In *ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pages 83–86, 2015.

[24] J. Ou, B. Yu, and D. Z. Pan. Concurrent guiding template assignment and redundant via insertion for DSA-MP hybrid lithography. In *ACM International Symposium on Physical Design (ISPD)*, pages 39–46, 2016.

[25] J. Ryckaert, P. Raghavan, P. Schuddinck, H. B. Trong, A. Mallik, S. S. Sakhare, B. Chava, Y. Sherazi, P. Leray, A. Mercha, et al. DTCO at N7 and beyond: patterning and electrical compromises and opportunities. In *Proceedings of SPIE*, volume 9427, 2015.

[26] Y. Seino, H. Yonemitsu, H. Sato, M. Kanno, H. Kato, K. Kobayashi, A. Kawanishi, T. Azuma, M. Muramatsu, S. Nagahara, et al. Contact hole shrink process using directed self-assembly. In *Proceedings of SPIE*, volume 8323, 2012.

[27] S. Shim, W. Chung, and Y. Shin. Defect probability of directed self-assembly lithography: Fast identification and post-placement optimization. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 404–409, 2015.

[28] H.-S. P. Wong, C. Bencher, H. Yi, X.-Y. Bao, and L.-W. Chang. Block copolymer directed self-assembly enables sublithographic patterning for device fabrication. In *Proceedings of SPIE*, volume 8323, 2012.

[29] S. Wuister, T. Druzhinina, D. Ambesi, B. Laenens, L. H. Yi, and J. Finders. Influence of litho patterning on DSA placement errors. In *Proceedings of SPIE*, volume 9049, 2014.

[30] Z. Xiao, Y. Du, M. D. F. Wong, and H. Zhang. DSA template mask determination and cut redistribution for advanced 1D gridded design. In *Proceedings of SPIE*, volume 8880, 2013.

[31] Z. Xiao, C.-X. Lin, M. D. F. Wong, and H. Zhang. Contact layer decomposition to enable DSA with multi-patterning technique for standard cell based layout. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 95–102, 2016.

[32] H. Yi, X.-Y. Bao, J. Zhang, R. Tiberio, J. Conway, L.-W. Chang, S. Mitra, and H.-S. P. Wong. Contact-hole patterning for random logic circuit using block copolymer directed self-assembly. In *Proceedings of SPIE*, volume 8323, 2012.

[33] B. Yu, X. Xu, S. Roy, Y. Lin, J. Ou, and D. Z. Pan. Design for manufacturability and reliability in extreme scaling vlsi. *SCIENCE CHINA Information Science*, 2016.