Technology path-finding framework for directed-self assembly for via layers

Yasmine Badr
Puneet Gupta

Technology path-finding framework for directed-self assembly for via layers

Yasmine Badr* and **Puneet Gupta**
University of California, Electrical Engineering Department, Los Angeles, California, United States

Abstract. Directed self assembly (DSA) is a very promising patterning technology for the sub-7-nm technology nodes, especially for via/contact layers. In the graphoepitaxy type of DSA, a complementary lithography technique is used to print the guiding templates, where the block copolymer (BCP) phase-separates into regular structures. Accordingly, the design-friendliness of a DSA-based technology is affected by several factors: the complementary lithography technique, the legal guiding templates, the number of masks/exposures used to print the templates, the related design rules, the forbidden patterns (hotspots), and the characteristics of the BCP. Thus, foundries have a huge number of choices to make for a future DSA-based technology, affecting the design-friendliness, and the cost of the technology. We propose a framework for DSA technology path-finding, for via layers, to be used by the foundry as part of design and technology co-optimization. The framework optimally evaluates a DSA-based technology in which an arbitrary lithography technique is used to print the guiding templates, possibly using many masks/exposures, and provides a design-friendliness metric. In addition, if the evaluated technology is not design-friendly, the framework computes the minimum-cost technology change that makes the technology design-friendly. The framework is used to evaluate technologies like DSA+193-nm immersion (193i) lithography, DSA+extreme ultraviolet (EUV), and DSA+193i self-aligned double patterning. For example, one study showed that one mask of EUV in a DSA+EUV technology can replace three masks of 193i in a DSA+193i technology. © *2017 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.JMM.16.1.013505]

Keywords: directed self assembly; multiple patterning; design and technology co-optimization; technology path-finding; 5 nm; Moore's law.

Paper 17001P received Jan. 11, 2017; accepted for publication Mar. 14, 2017; published online Mar. 28, 2017.

1 Introduction

Directed self assembly (DSA) is a promising patterning technique for the sub-7-nm nodes because of its inherent pitch multiplication features and low cost,[1] especially for via layers. There are two main types of DSA: graphoepitaxy and chemoepitaxy, but we focus on graphoepitaxy because it is more appropriate for patterning of random features, and this is required for via/contact layers.[2] A diagram showing graphoepitaxy is shown in Fig. 1. First, the guiding templates are defined using a lithography technique, then the block copolymer (BCP) undergoes annealing and self-assembles into regular structures (cylindrical formations in this example). This way, via holes with a pitch smaller than that allowed by the lithography technique can be realized, as shown in the case of the two neighboring holes in the same guiding template in Fig. 1.

A DSA technology for via layers is characterized by a lot of factors. First, a complementary lithography technique is needed in graphoepitaxy to print the guiding templates. The candidate complementary lithography techniques include 193-nm immersion lithography (193i), extreme ultraviolet (EUV) lithography, E-beam direct write, self-aligned double patterning (SADP) as well as possibly any other emergent technology. The choice of the complementary lithography technique will determine the legal guiding templates. The legal templates, along with the BCP properties, determine the legal DSA groups, where a DSA group is a set of vias that are to be patterned in the same guiding template, as is the case for the two vias sharing the same template in Fig. 1. For example if EUV or E-beam is used, then the templates for more complicated DSA groups (e.g., L-shaped groups) may be printed, whereas if 193i is used then only collinear groups are allowed, as shown in Fig. 2, due to the higher lithography variations in the case of 193i which leads to higher defectivity in self-assembly.[3] The BCP properties also determine the allowed contact/via pitches that can be manufactured by self-assembly. Moreover, the guiding templates may be patterned using several masks/exposures [multiple patterning (MP)]. Finally, if the foundry has a database of hotspots (forbidden patterns), it is required to prohibit DSA groups that will lead to templates causing any of the hotspots. These factors need to be evaluated during the technology path-finding of future nodes using DSA.

We propose a DSA technology exploration framework for via layers. The input to the framework is the specifications of the technology to be explored and a benchmark layout. The framework evaluates the technology, from the point of view of design compliance and provides a design friendliness metric. If the technology is not design-friendly, then the framework computes the minimum-cost change to the technology that would make it compliant to the provided benchmark design. The objective of this framework is to be used by the foundry for design and technology co-optimization (DTCO) in order to develop a new technology node, and not for processing large full chip-layouts for technologies already in production.

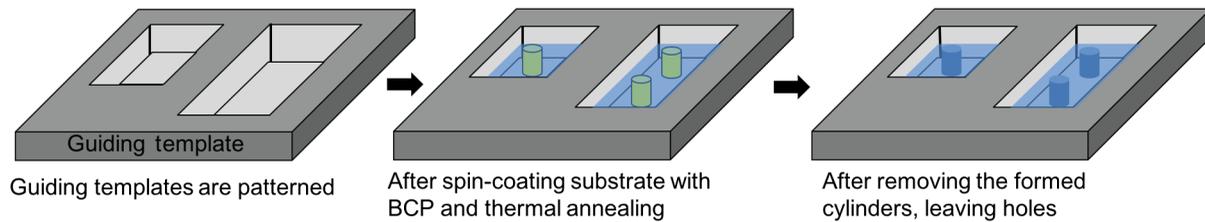*Address all correspondence to: Yasmine Badr, Email: ybadr@ucla.edu

Fig. 1 DSA process for contact/via holes.

The contribution of this work can be summarized as follows:

1. To the best of our knowledge, this is the first optimal and general framework to be proposed for evaluation of any DSA-based technology (using any complementary lithography technique) that can have multiple masks/exposures to print the guiding templates. (By optimal evaluation, we mean that given that the assumptions employed in modeling the problem are reasonable and justifiable, the mathematical formulation is solved using an optimal solver, and not using heuristics.)

2. Our framework manifests correct-by-construction methods to avoid DSA templates that create technology-specific hotspots.

3. If the evaluated technology is not design-friendly, the framework computes the minimum-cost technology change that makes the technology design-friendly.

4. Several novel technologies are evaluated using the proposed framework, including DSA+EUV, DSA+SADP, and DSA+E-beam.

The rest of this paper is organized as follows: Sec. 2 discusses the related work in the literature. Section 3 presents an overview of the framework. Section 4 breaks down the framework into a sequence of stages, and describes them in detail. The integer linear program (ILP) formulation for path-finding is presented in Sec. 5. Section 6 describes the minimum-cost technology change computation. In Sec. 7, we show case studies that have been performed using the framework, followed by conclusion and future work in Sec. 8.

2 Prior Work

The need for the co-optimization of BCP, design, and lithography in order to find a design-friendly technology with lithography-friendly guiding templates, using the minimum
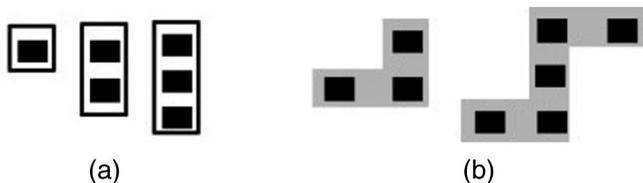

Fig. 2 Examples showing that the lithography technique used to print DSA guiding templates affects the allowed DSA groups in (a) both 193i and EUV and (b) EUV not in 193i.

number of mask/exposures, is emphasized by Ma et al.[4] However, their work focuses on the selection of the BCP and the guiding template dimensions to maximize the robustness in self-assembly, and they do not offer methods for optimizing the technology for design-friendliness. There is a lot of research targeting the optimization and verification of the guiding templates in order to generate the required self-assembled shapes. Approaches in this category have used the combinations of simulation and mathematical models as in the work of Ma et al.,[5] machine learning as in the work of Xiao et al.,[6] level-set based algorithm with self-consistent field theory in the work of Ouaknin et al.,[7] in addition to the experimental studies performed by Gharbi et al.[1] Our framework is not to be used for the purpose of optimizing the templates for the robustness of the self-assembly process, but it is used to determine the DSA groups that are important for the design; generating the actual guiding template shapes is not within the scope of this framework.

Another category of research aims at achieving DSA-friendly design. For example, the design of a DSA-compliant contact layer in standard cells has been studied by Du et al.,[8] assuming single patterning (SP) of the guiding templates. Yi et al.[9] showed a design strategy (no automated design methods) for standard cell design based on the requirements of DSA technology, so it cannot be used to choose a design-friendly technology. DSA-aware routing has been addressed by Du et al.[10] Shim et al.[11] proposed a method for perturbing the placement of standard cells, in order to decrease the DSA defect probability. In addition, the traditional idea of dummy via insertion has been revived in the works of Fang et al.[12] and Ou et al.,[13] with the new objective of DSA-compliance. The work of Lin and Chang[14,15] develops a cut redistribution algorithm to be able to print cuts in gridded layouts using DSA. The work of Wang et al.[16] can find non-DSA-friendly configurations by finding the configurations that result in defective self-assembly through simulation. The third category of research develops algorithms for hybrid technologies involving DSA, DSA+EUV, and DSA+MP for 193i. Several works[17–20] perform DSA-aware mask assignment for DSA+193i technology. In addition, Ou et al.[13] solved the same problem while adding redundant vias, while Lin et al.[15] added cut redistribution. Karageorgos et al.[21] solved the same problem with a variable number of masks, but can only run on a cluster of 15 vias at most, using exhaustive enumeration of grouping and mask assignment options, then they extend the work[22] to employ heuristics for bigger clusters of vias, potentially sacrificing optimality. Gronheid et al.[2] used experimental work to show advantages of using DSA+EUV. None of the works on hybrid DSA technologies offers the capability of modeling different and arbitrary DSA technologies optimally on a macrolayout.
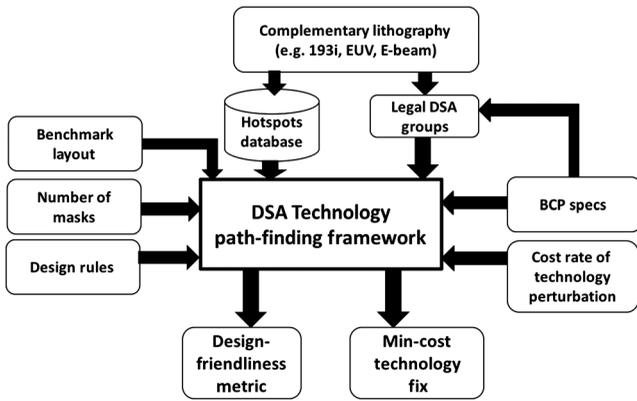
Fig. 3 Overview of the hybrid DSA technology exploration framework.

3 Overview of the Framework

The overview of the framework is shown in Fig. 3. The framework takes as input the specifications of the technology under evaluation, which are the following:

BCP Specifications. These are the minimum pitch to which the BCP can be compressed (min_dsa_pitch), the maximum pitch to which the BCP can be stretched (max_dsa_pitch), assembled holes dimension (via_width), and the maximum allowed number of vias per DSA group (max_g). [note that the natural pitch (L_0) of the BCP lies between min_dsa_pitch and max_dsa_pitch]. The max_g constraint exists because earlier research has shown that smaller DSA group sizes can lead to more robust self-assembly.[23]

Number of masks. This is the number of masks/exposures used to print the guiding templates, in case of MP.

Design rules. These include min_pitch_same_mask which is the minimum allowed pitch on a mask, and min_pitch_diff_mask which is the minimum allowed pitch between any two guiding templates even if they are assigned to different masks. (If the technology under evaluation assumes that self-assembly is done one time only after all litho etch steps [i.e., $(litho - etch)^x + DSA$], then the min_pitch_diff_mask rule should be satisfied even if the shapes are assigned to different masks because of overlay error. However, if the assumed process performs self-assembly after each litho-etch step [i.e., $(litho - etch - DSA - etch)^x$], then this rule is not needed and should be set to zero.) Unidirectionality of the masks can also be enforced dictating that the shapes on each mask should all be in the same direction (vertical/horizontal).

Specifications of the legal DSA groups. These are the properties of the allowed DSA groups. Properties include manhattan only, collinear only, and equidistant vias only (i.e., pairwise distances between neighboring vias in a DSA group should be identical). Since the proposed framework is intended to be capable of modeling any arbitrary technology, the framework provides the option of defining a custom legal grouping checker, which has properties different from the options that are already provided, and this is done by implementing a well-defined and simple interface for the grouping checker and using the framework in an application programming interface (API)-like fashion.

Hotspots database. These are the patterns that are forbidden by the technology under evaluation. More details are provided in Sec. 4.5.

Table 1 Definition of input parameters.

Parameter	Definition
min_dsa_pitch	Minimum pitch to which the BCP can be compressed
max_dsa_pitch	Maximum pitch to which the BCP can be stretched
max_g	Maximum allowed number of vias per DSA group
via_width	Width of the via hole
min_pitch_same_mask	Minimum allowed pitch on a mask
min_pitch_diff_mask	Minimum allowed pitch between any two guiding templates even if they are assigned to different masks

Cost rate of technology perturbation. These are the costs of changing the technology. These costs are used by the framework to compute the minimum-cost perturbation to the technology that can make it design-compliant (more details are in Sec. 6).

The definitions of the input parameters are summarized in Table 1.

The output of the framework is a design-friendliness metric for the technology, which is the number of violations on the used benchmark. In addition, the framework shows the resulting DSA groups for each mask/exposure.

To the best of our knowledge, this is the first DTCO framework that can be used to optimally evaluate any DSA-based technology. (The framework is available for download from Ref. 24.) The reasons for this generalization capability are: first, the foundry can define and use its own grouping checker with custom/nonconventional allowed or restricted groups through the API. Second, a hotspots database is used as input to the tool, in order to model any forbidden via configurations that should not be allowed.

4 Components of the Directed-Self Assembly Path-Finding Framework

The flow of the proposed framework is shown in Fig. 4. First, the candidate DSA groups are generated. Then, the pairs of groups that cannot coexist are determined. After that, the group combinations that will result in a forbidden pattern
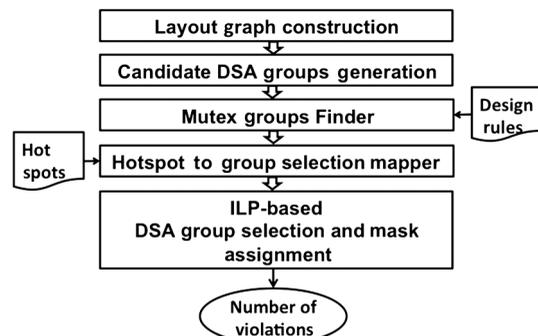

Fig. 4 Flow of the DSA path-finding framework.

if assigned to the same mask are found. Finally, the output of the previous steps is used to formulate and solve an ILP that simultaneously performs the group selection and assigns the selected DSA groups including singletons to the masks. (A singleton is a DSA group containing one via only.)

4.1 Layout Graph Construction

Given the via layer, a graph is constructed such that a graph node is created for each via and a graph edge exists between any two vias whose center-to-center distance is less than min_pitch_same_mask. This step runs in $O(n)$, where n is the number of vias.
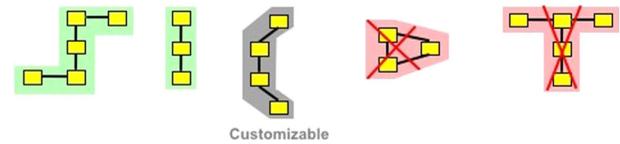
4.2 Candidate Group Generation

All the candidate legal grouping options are generated in this step, starting at each graph node. This is performed on two stages:

1. *Finding grouping options*: Starting at a particular graph node, the layout graph is used to find all the possible strongly connected subgraphs that contain this node and with a number of nodes less than or equal to max_g. This is done by a custom graph traversal algorithm, which saves such subgraphs. This traversal truncates the search from each subgraph as soon as it contains max_g nodes. Practically, the number of strongly connected subgraphs is not very large due to the constraint that, for each node, we enumerate only the subgraphs having a number of nodes less than or equal to max_g. Assuming max_g has a small value, which is usually the case due to DSA yield issues,[23] this process runs in $O(n)$, where n is the number of vias. The analysis of this complexity is as follows. Assume the average branching factor (number of neighbors of a node) of the graph is b. The desired maximum number of nodes in the strongly connected subgraph is max_g. Then, we enumerate the strongly connected subgraphs by finding all paths of depth max_g or smaller, starting at each node in the graph. So starting at a particular node, the number of these paths is $O(b^0 + b^1 + b^2 + \cdots + b^{\max\_G}) = O(b^{\max\_G+1})$. Since we enumerate these paths starting at each node, then the total number of strongly connected subgraphs is $O(n * b^{\max\_G+1})$. In the case of our via graphs, b is the number of vias within min_pitch_same_mask from a particular via, and this number is usually small; and with max_G usually being a very small number (~3), the complexity becomes $O(n)$.

2. *Finding candidate groups*: Not all the grouping options are valid DSA groups. Thus, a technology-specific grouping checker is run on each grouping option, in order to disqualify the noncompliant ones. Grouping checkers are explained in Sec. 4.3.

4.3 Grouping Checkers

The specific requirements of the technology are modeled in the technology-specific grouping checker used by the framework. Some common checkers are provided, such as the collinear grouping checker typically used for 193i and the more flexible grouping checker, which is used for EUV

Fig. 5 Examples of legal and Illegal groups according to our EUV grouping checker. A line between two vias means that distance between their centers is less than min_pitch_same_mask.

experiments. Other options are also provided, such as requiring all neighboring vias inside the same group to be equidistant. However, any different grouping checker, which is very specific to any arbitrary technology under evaluation, can also be customized through the API exposed by the framework. Two examples of grouping checkers are presented next.

4.3.1 193i grouping checker

In the 193i experiments, a manhattan and collinear grouping checker is used. Given a grouping option represented as a set of vias, the checker considers a group legal if all centers of the vias are vertically or horizontally aligned (all vias must be square shapes with a dimension equal to the assembled hole diameter) and the center-to-center distance between every two neighboring vias in the group is within the allowed BCP pitch range.

4.3.2 EUV grouping checker

In the EUV experiments, it has been assumed that the legal group can be any nonself-intersecting chain of vias. The following groups are illegal:

1. Groups whose graphs, similar to the graph explained in Sec. 4.1, have a cycle. This is because such groups will require donut-shape templates in order to confine the self assembly process, and such templates have been assumed difficult to print.

2. Groups whose graphs have T-shapes or Fork structures, since it has been assumed that the self-assembly in such a configuration has high defectivity due to the existence of many corners leading to lithography variation and lack of strong confinement[3] (unless high-NA EUV is in use, then such restriction can be alleviated).

In addition, the distance between every two neighboring vias must satisfy the BCP pitch range. Figure 5 shows some examples of EUV groups that are legal in green, others that are illegal in red, and a nonmanhattan group, which can be determined legal or illegal according to the used knob allowing or disallowing nonmanhattan neighborhood.

4.4 Mutually Exclusive (MUTEX) Groups Finder

A set of two or more DSA groups may not be allowed to coexist even though each of them is a legal DSA group. This can happen in the following cases:

MUTEX case 1. A set of groups has one (or more) common via(s). Out of all the candidate groups involving a certain via, only one group can be selected. An example is shown in Fig. 6(a).
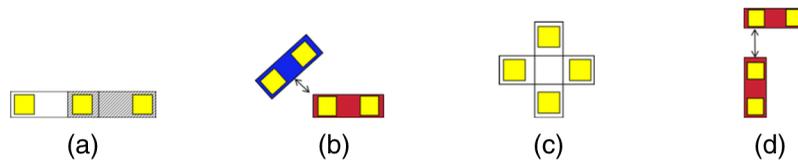
Fig. 6 Examples showing the four MUTEX groups cases. (a)–(d) MUTEX cases 1 to 4.

MUTEX case 2. Distance between two groups is smaller than (min_pitch_diff_mask−via_width). Thus, only one of these groups can be selected (see the definition of the used rules in Sec. 3). An example is shown in Fig. 6(b). Note that if one of the two groups in MUTEX case 2 is a singleton (i.e., nongrouped via), the nonsingleton group is removed from the grouping options. This is because the nonsingleton group will result in a design rule violation, regardless of which mask it gets assigned to, even though the input via layer is DRC-clean.

MUTEX case 3. Two groups overlap geometrically. However, some processes may allow the groups to overlap if they are assigned to different masks. (This can be done if self-assembly is done for each mask then the assembled holes are transferred to a hard mask.) Thus, the input knobs of the framework can disable this case. An example is shown in Fig. 6(c).

MUTEX case 4. Distance between two groups is smaller than (min_pitch_same_mask−via_width). The two groups can be selected only if they are assigned to different masks. An example is shown in Fig. 6(d).

To find the MUTEX groups belonging to cases 2 to 4 described above, the neighborhood of each via is examined in order to find such pairs of groups. For a quad-tree implementation of region query, finding MUTEX groups runs in $O(n^{1.5})$. MUTEX groups are fed into the ILP formulation (Sec. 5.3).

4.5 Hot Spot to Group Selection Mapper

In addition to the MUTEX groups described in Sec. 4.4, some groups cannot be assigned to the same mask because they will cause hotspots, even though they satisfy the design rules. A hotspot can be one of the following:

1. *Lithographic hotspot.* This is a low-yield pattern, which is likely to cause a printing failure.[25]
2. *Complex design rule.* In advanced nodes, foundries had to introduce a lot of complex 2-D and conditional rules. These rules can require pattern-based representation.[26]
3. Forbidden pattern due to the use of a restrictive patterning technology like self-aligned MP.

Thus in order to have a correct evaluation for the technology, the generated DSA templates must be hotspot-free. Moreover, using forbidden patterns, the framework can be used to model and evaluate any new technology with unusual pattern-based requirements.

We use the same pattern representation proposed by Badr et al.,[27] in which the segment representation is used to encode
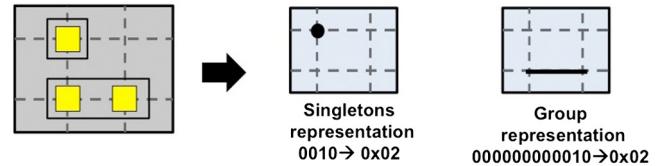

Fig. 7 A hotspot and its corresponding representation.

the groups of size two or bigger and the node representation is used to encode the singletons. Both representations are needed for every hotspot. For example, Fig. 7 shows an example of a 2×2 hotspot and its group and singleton representation, where the nodes and segments are written as a binary string and stored as the equivalent number. Only gridded layouts can have hotspots, in this framework.

A hotspot on a mask is defined by a list of segments that are occupied by DSA groups, a list of segments that are empty, a list of nodes that are filled due to singletons, and a list of nodes that are empty (i.e., no singletons exist at the node location).

The framework performs the grouping and mask assignment such that none of the hotspots occurs in any window in the mask. This is done by scanning all nonempty windows and generating the forbidden combinations of groups. This is performed in $O(k * n)$, where k is the number of hotspots and n is the number of vias. For each hotspot and for each nonempty layout window, the following sets of groups are defined:
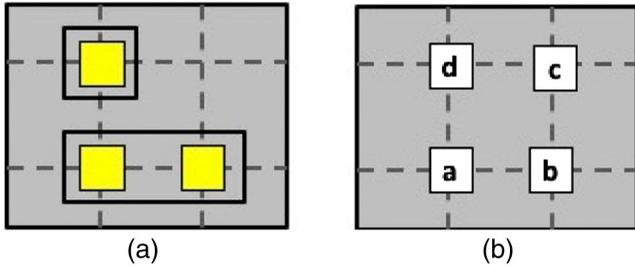
On groups: Groups that need to exist in order to form the hotspot. For every filled segment in the hotspot pattern, one group that spans the segment must be on. Since one segment can be filled by one of several candidate groups, there can be several sets of on groups for a certain window and for a certain hotspot.

Off groups: Groups that need to be absent in order to create the hotspot. For every empty segment in the pattern, all the groups that span it must be off. In addition, for every occupied node, all the candidate groups of size bigger than one for the via at this location (in the window) must be off [i.e., the via at this location (if any) in the window must exist as a singleton].

Absent singletons: Vias that need to be absent in order to form the hotspot. For every empty node; the via existing at this location (if any) in the window must not exist as a singleton.

The forbidden group combinations will then be used in the ILP.

For example, for the hypothetical hotspot shown in Fig. 8(a) to exist in the layout window shown in Fig. 8(b), there is only one set of on groups in this case and it is $\{g_{\{a,b\}}\}$, the set of off groups is $\{g_{\{b,c\}}, g_{\{c,d\}}, g_{\{a,d\}}\}$, and the set of absent singletons is $\{c\}$.

Fig. 8 Example showing the different sets of groups generated for one hypothetical (a) hotspot and one (b) layout window. In this example, on groups: $\{g_{\{a,b\}}\}$, off groups: $\{g_{\{b,c\}}, g_{\{c,d\}}, g_{\{a,d\}}\}$, absent singletons: {c}.

5 Path-Finding Solution Using ILP

An ILP is used to do the group selection and mask assignment for the selected groups, simultaneously. (In the case of evaluating a single exposure technology, the same formulation is used but there will be no mask or similarity variables, so the result is only the group selection.)

The used constraints are derived from the input to the framework described in Sec. 3, the candidate DSA groups as explained in Sec. 4.2, the MUTEX groups as described in Sec. 4.4 as well as the forbidden group combinations due to hotspots as explained in Sec. 4.5.

A conflict exists between two vias if their center-to-center distance is less than min_pitch_same_mask; they are assigned to the same mask and are not in the same DSA group.

The variables and notation used are explained in Table 2.

Although the ILP works for one mask (SP), two masks [double patterning (DP)], three masks [triple patterning (TP)], or four masks [quadruple patterning (QP)] or any higher power of two, the mathematical formulation is presented assuming four masks for the sake of simplicity of the notation. (In the case of TP, other constraints are added in order to prohibit the unused mask bit combinations, similar to the work of Yu et al.[28] and Badr et al.[18] Similarly, in order to support the quintuple, sextuple patterning, or other number of masks that is not a whole power of two, constraints are needed to prevent the unused mask bit combinations.)

5.1 Objective Function

The objective function in Eq. (1) aims at minimizing the number of conflicts. As explained earlier, a conflict exists between two vias if there is a graph edge between them and they are not in the same DSA group. Note that the objective function does not differentiate between two solutions of different group selections, as long as both solutions have the same number of conflicts. However, the cost function can be modified to add a weighted sum for the groups, and to add a weight to the sum of conflicts. The weight for each group should be inversely proportional to the expected yield of the group, and the sum of the weights of the chosen groups must still be lower than the weight of one conflict. However, with the absence of a succinct but accurate yield model, the notion of optimality of the solution will be suspect, and thus we avoid the weighted groups idea in the cost function

$$\text{minimize} \sum_i \sum_j c_{ij}. \tag{1}$$

Table 2 Notation used in ILP formulation.

c_{ij}	Variable indicating if i'th and j'th vias are in conflict
m_i^b	b'th bit of mask variable of i'th via
s_{ij}^b	Similarity variable indicating if b'th bit in mask of i'th via is identical to the b'th bit in mask of j'th via
GEs	Set of graph edges in the layout graph
$\mathbf{P}_k$	k'th set of vias which can form a legal group
K	Number of candidate groups
$g_\mathbf{I}$	Flag indicating if the vias in set $\mathbf{I}$ are grouped. Variable only exists if the vias in set $\mathbf{I}$ form a candidate group and if $|\mathbf{I}| \geq 2$
$\text{dir}(\mathbf{P}_k)$	Orientation of the candidate DSA group formed of $\mathbf{P}_k$. Value is v if vertical; h if horizontal; o if noncollinear.
$\mathbf{E}_m$	m'th set of MUTEX DSA groups
M	Number of sets of MUTEX DSA groups of MUTEX cases 1 to 3
N	Number of sets of MUTEX DSA groups of MUTEX case 4
	Notation for the hot spot prevention constraints
ONG_{wq}^h	q'th set of ON groups for the h'th hotspot, for the w'th layout window
OFFG_w^h	Set of OFF groups for the h'th hotspot, for the w'th layout window
AS_w^h	Set of absent singletons for the h'th hotspot, for the w'th layout window
n_{wq}^h	Index of an arbitrary via in an arbitrary group in ONG_{wq}^h
f_w^h	Index of an arbitrary via in an arbitrary group in OFFG_w^h
$\mathbf{N}_{wq}^{hy}$	y'th group in ONG_{wq}^h
$\mathbf{F}_w^{hy}$	y'th group in OFFG_w^h

5.2 Constraints Between Vias

Constraints are added to assert the conflict variable between two vias if there is a graph edge between them; they are assigned to the same mask and none of the grouping options including both vias is asserted, as shown in Eq. (2). To represent the problem in linear constraints, binary variables are used to encode the mask number, like the work of Yu et al.[28] The constraints in Eqs. (3)–(10) are used to assert the similarity variable between any two vias if they are assigned to the same mask, i.e., they force the similarity variable to be the output of XNOR between the two corresponding mask bits. For example, for a pair of vias i and j, if they are both assigned to mask 3, then $m_i^1 = m_j^1 = 1$ and $m_i^2 = m_j^2 = 1$. Accordingly, the similarity variables s_{ij}^1 and s_{ij}^2 are both

set to 1 due to Eqs. (3)–(10). In addition, the constraints in Eqs. (11) and (12) are added in order to allow the selection of the DSA group only if all the involved vias are assigned to the same mask.

$$s_{ij}^1 + s_{ij}^2 - \sum_{\substack{k \in [1..K] \\ \{i,j\} \subseteq \mathbf{P}_k}} g_{\mathbf{P}_k} \leq c_{ij} + 1 \quad \forall\, (i,j) \in \text{GEs}, \tag{2}$$

$$s_{ij}^1 \geq 1 - m_i^1 - m_j^1 \quad \forall\, (i,j) \in \text{GEs}, \tag{3}$$

$$s_{ij}^1 \leq 1 - m_i^1 + m_j^1 \quad \forall\, (i,j) \in \text{GEs}, \tag{4}$$

$$s_{ij}^1 \leq 1 + m_i^1 - m_j^1 \quad \forall\, (i,j) \in \text{GEs}, \tag{5}$$

$$s_{ij}^1 \geq -1 + m_i^1 + m_j^1 \quad \forall\, (i,j) \in \text{GEs}, \tag{6}$$

$$s_{ij}^2 \geq 1 - m_i^2 - m_j^2 \quad \forall\, (i,j) \in \text{GEs}, \tag{7}$$

$$s_{ij}^2 \leq 1 - m_i^2 + m_j^2 \quad \forall\, (i,j) \in \text{GEs}, \tag{8}$$

$$s_{ij}^2 \leq 1 + m_i^2 - m_j^2 \quad \forall\, (i,j) \in \text{GEs}, \tag{9}$$

$$s_{ij}^2 \geq -1 + m_i^2 + m_j^2 \quad \forall\, (i,j) \in \text{GEs}, \tag{10}$$

$$s_{ij}^1 \geq g_{\mathbf{P}_k} \quad \forall\, \{i,j,k | (i,j) \in \text{GEs}, \{i,j\} \subseteq \mathbf{P}_k, k \in [1\ldots K]\}, \tag{11}$$

$$s_{ij}^2 \geq g_{\mathbf{P}_k} \quad \forall\, \{i,j,k | (i,j) \in \text{GEs}, \{i,j\} \subseteq \mathbf{P}_k, k \in [1\ldots K]\}. \tag{12}$$

5.3 Mutual Exclusive Group Constraints

As explained in Sec. 4.4, some groups cannot coexist due to MUTEX cases 1 to 4.

5.3.1 Constraints for MUTEX cases 1 to 3

For MUTEX cases 1 to 3, constraints in Eq. (13) are generated to prohibit the selection of more than one group from each set of MUTEX groups

$$\sum_{g \in \mathbf{E_i}} g \leq 1 \quad \forall\, i \in [1\ldots M]. \tag{13}$$

5.3.2 Constraints for MUTEX case 4

For MUTEX case 4, two mutually exclusive groups can coexist only if they are assigned to different masks. The constraints in Eqs. (14) and (15) prevent every pair of MUTEX groups of case 4 from being assigned to the same mask if they are both selected, in the case of the two groups being nonsingletons and in the case of one of the two groups being a singleton, respectively

$$g_A + g_B + s_{xy}^1 + s_{xy}^2 \leq 3 \quad \forall\, n \in [1\ldots N]$$
$$\text{s.t. } \mathbf{E}_n = \{\mathbf{A}, \mathbf{B}\}, \quad |\mathbf{A}| \geq 2, \quad |\mathbf{B}| \geq 2, \quad x \in \mathbf{A}, \quad y \in \mathbf{B}, \tag{14}$$

$$g_A + s_{xy}^1 + s_{xy}^2 \leq 2 \quad \forall\, n \in [1\ldots N]$$
$$\text{s.t. } \mathbf{E}_n = \{\mathbf{A}, \mathbf{B}\}, \quad |\mathbf{A}| \geq 2, \quad x \in \mathbf{A}, \mathbf{B} = \{y\}. \tag{15}$$

5.4 Hotspot Prevention Constraints

Constraints are added in order to prevent the existence of guiding templates that create hotspots. As explained in Sec. 4.5, for each hotspot pattern and a layout window, there is one or more sets of on groups, a set of off groups, and a set of absent singletons. Thus, the constraints in Eq. (16) are generated in order to prevent at least one of the required conditions for a hotspot from occurring on any mask. That is, at least one of the on groups is not selected or is not on the same mask as the rest, or one of the off groups is selected and assigned to the same mask, or one of the absent singletons is present on the same mask. The similarity variables between the vias are used along with the grouping variables (see Table 2) to enforce that

$$\sum_{\mathbf{A} \in \text{ONG}_{wq}^h} g_{\mathbf{A}} + \sum_{\mathbf{B} \in \text{OFFG}_w^h} (1 - g_{\mathbf{B}}) + \sum_{k=1}^{k=2} \sum_{\substack{x \in \text{AS}_w^h \\ x \neq (n_{wq}^h)}} [1 - s_{x(n_{wq}^h)}^k]$$

$$+ \sum_{k=1}^{k=2} \sum_{\substack{y=1 \\ i \in \mathbf{N}_{wq}^{hy} \\ j \in \mathbf{N}_{wq}^{h(y+1)} \\ i \neq j}}^{y=|\text{ONG}_{wq}^h|-1} s_{ij}^k + \sum_{k=1}^{k=2} \sum_{\substack{z=1 \\ i \in \mathbf{F}_w^{hz} \\ j \in \mathbf{F}_w^{h(z+1)} \\ i \neq j}}^{z=|\text{OFFG}_w^h|-1} s_{ij}^k + \sum_{k=1}^{k=2} s_{(n_{wq}^h)(f_w^h)}$$

$$\leq 3(|\text{ONG}_{wq}^h| + |\text{OFFG}_w^h|) + 2|\text{AS}_w^h| - 3 \quad \forall\, q, h, w. \tag{16}$$

5.5 Unidirectional Group Constraints

Unidirectional layers have become favorable in advanced nodes using 193i in order to make the most benefit of polarized illumination and off-axis Illumination.[29] The framework provides the option to force the formed groups on each mask to follow a certain orientation (vertical or horizontal). For unidirectional masks with QP, two masks are vertical and the other two masks are horizontal; for TP, one mask is vertical and the other two are horizontal or vice versa; finally for DP, one mask is vertical and the other is horizontal. For QP, the constraints in Eqs. (17) and (18) force each vertical group to be assigned to mask 1 or mask 2 if the group is selected, and each horizontal group to be assigned to mask 3 or mask 4 if the group is selected. A vertical group is a group of two or more vias that are aligned on the same Y-axis, whereas a horizontal group is a group of two or more vias that are aligned on the same X-axis. Singletons are not constrained to any direction because the template for a singleton is likely to have aspect ratio of 1:1 (Ref. 1)

$$g_{\mathbf{P}_k} + m_i^1 \leq 1 \quad \forall\, \{k, i | k \in [1\ldots K], i \in \mathbf{P}_k, \text{dir}(\mathbf{P}_k) = \text{`}v\text{'}\}, \tag{17}$$

$$g_{\mathbf{P}_k} - m_i^1 \leq 0 \quad \forall \{k, i | k \in [1 \ldots K], i \in \mathbf{P}_k, \operatorname{dir}(\mathbf{P}_k) = `h'\}.$$

$$(18)$$

5.6 Parallelization

It is required to solve the ILP in parallel in order to reduce runtime without sacrificing optimality. Thus, the connected components[30] of the graph are determined, and the ILP for each connected component is constructed and solved independently. Multiple threads are used to solve the ILPs for the components.

6 Minimum-Cost Technology Fix

If the number of violations is not zero, meaning that the technology is not design-friendly, we propose to find the minimum change to the technology rules to ensure design compliance. Note that we also allow some design rule value changes that may require design fixes as well. Any DSA-aware design flows that exist are captured in the design benchmarks used. The technology parameters that are allowed to change are the following:

Decreasing: min_pitch_same_mask which means allowing a smaller distance on the same mask and may translate to additional resolution enhancement cost. A cost rate of changing this value by 1 nm is one of the inputs to the framework.

Decreasing: min_pitch_diff_mask which means allowing a smaller distance between any two DSA groups on different masks and may translate to additional costs in overlay control and etch.

Increasing: max_dsa_pitch which indicates that the BCP can be stretched to a longer distance and translates to costs in resolution enhancement of templates and masks to get better confinement or BCP optimization.

For the above three parameters, cost is expressed per nm change.

Increasing: max_g, allowing larger DSA group sizes, which translates to BCP optimization and/or better confinement via enhanced resolution in printing templates.

Removing a specific hotspot: Each hotspot in the input hotspots database can be removed, which can be achieved by forcing a design change or a patterning/OPC change.

Removing the unidirectionality constraint: On the DSA groups of each mask which again can require more aggressive, expensive OPC, or incur yield loss.

Allowing DSA groups on different masks to overlap. The constraint that prevents the existence of two geometrically overlapping DSA groups on two different masks can be relaxed. This comes at the cost of needing multiple self-assembly steps in the process instead of one.

Using an alternative grouping checker. The alternative grouping checker that allows grouping of any set of vias is similar to the EUV grouping checker explained in Sec. 4.2, with nonmanhattan groups allowed. This would usually mean using a different, more expensive template patterning scheme.

An ILP is formulated in order to find the minimum-cost change to make the technology design-friendly.

6.1 Technology Change ILP Formulation

This ILP is a variant of the one explained in Sec. 5. The added notation used in the ILP is shown in Table 3, showing

Table 3 Notation used for minimum-cost technology change ILP.

e_sm_d	Variable indicating if min_pitch_same_mask is relaxed to d
e_dm_d	Variable indicating if min_pitch_diff_mask is relaxed to d
e_h_i	Variable indicating if the min-cost technology change involves removing the i'th hotspot
e_u	Variable indicating if the min-cost technology change involves removing the unidirectionality constraint
eov	Variable indicating if the overlap between templates on different masks is part of the min-cost technology change
$e\_gc_x^d$	Variable indicating if the min-cost technology change involves a relaxed grouping condition allowed by the original grouping checker under evaluation, max_g $= x$ and max_dsa_pitch $= d$
$e\_agc_x^d$	Variable indicating if the min-cost technology change involves a relaxed grouping condition allowed by the alternative grouping checker, max_g $= x$ and max_dsa_pitch $= d$
$GC(g_l)$	Set of grouping conditions (egc_x^d and $e\_agc_x^d$) allowing the grouping of set of vias l
$cost\_sm_d$	Cost of e_sm_d which is the input cost rate of changing min_pitch_same_mask multiplied by (d-min pitchsamemask). Default value of cost rate is 2/nm
$costdm_d$	Cost of e_dm_d which is the input cost rate of changing min_pitch_diff_mask multiplied by (d-min pitchdiffmask). Default value of cost rate is 2/nm
$costh_i$	Cost of eh_i (input). Default value is 1
$costu$	Cost of eu (input). Default value is 1
$costov$	Cost of eov (input). Default value is 4
$costgc_x^d$	Cost of $e\_gc_x^d$ which is the input cost rate of changing max_dsa_pitch multiplied by (d-max dsapitch)
$costagc_x^d$	Cost of $e\_agc_x^d$ which is the cost of the alternative checker added to the product of the input cost rate of changing max_dsa_pitch and (d-max dsapitch). Default value of cost of the alternative checker is 10
$dist_{ij}$	Distance between centers of vias i and j

the technology change variables as well as their associated costs. The technology change variables are similar to the idea of elastic programming[31] variables, except that in conventional elastic programming, a different elastic variable is added to each constraint that may need to be relaxed; but here the same technology change variable is added to all the constraints representing conflicts that will be resolved by the technology change.

The objective function is to minimize the cost of the selected technology changes, as shown in

$$\text{minimize} \sum_d \text{cost\_sm}_d * e\_\text{dm}_d + \sum_d \text{cost\_sm}_d * e_{\text{sm}}$$

$$+ \sum_i \text{cost\_h}_i * e\_h_i + \text{cost\_u} * e_u + \text{cost\_ov} * e\_\text{ov}$$

$$+ \sum_x \sum_d \text{cost\_gc}_x^d * e\_\text{gc}_x^d + \sum_x \sum_d \text{cost\_agc}_x^d * e\_\text{agc}_x^d. \tag{19}$$

The constraints in Sec. 5 have been modified in order to add the technology modifications. Constraints of Eq. (2) have been updated as shown in Eq. (20) where the technology change variables that would solve the conflict, represented by the constraint, are added. The conflict can be resolved by using a min_pitch_same_mask smaller than dist_{ij}, or using a bigger max_dsa_pitch or using the alternative grouping checker with same or bigger max_dsa_pitch. The conflict removal due to grouping is reflected in the addition of groups, which means that K has increased. Moreover, the conflict variables (c_{ij}) no longer exist, which means that the solution is not allowed to have any conflict/violation

$$s_{ij}^1 + s_{ij}^2 - \sum_{\substack{k \in [1..K] \\ \{i,j\} \subseteq \mathbf{P}_k}} g_{\mathbf{P}_k} \leq 1 + \sum_{d=1}^{d=\text{dist}_{ij}} e\_\text{sm}_d \quad \forall \, (i,j) \in \text{GEs}. \tag{20}$$

A constraint has been added to make sure that at most one grouping condition is selected as shown in

$$\sum_x \sum_d e\_\text{gc}_x^d + \sum_x \sum_d e\_\text{agc}_x^d \leq 1. \tag{21}$$

Another constraint has been added to pick a grouping condition variable enabling a certain group of vias, as shown in

$$\sum_{e \in \mathbf{P_k}(g_I)} e \geq g_{\mathbf{P}_k} \quad \forall \, \{k | k \in [1 \ldots K]\}. \tag{22}$$

The constraints for MUTEX cases 2 and 4 in Eq. (13) have been relaxed by adding the enabling change variables e_sm_d and e_dm_d, respectively, on the right-hand side (rhs) of the constraint. Similarly, the constraints for MUTEX case 3 in Eq. (13) have been relaxed by adding the variable that allows overlap (e_ov) between the groups on different masks on the rhs of the constraint. Similarly, the hotspot constraints in Eq. (16) have been relaxed by adding the corresponding hotspot removal variable e_h_i on the rhs.

7 Case Studies and Results

In this section, we present several exploration studies that have been done for DSA-based technologies using the proposed framework. The explored complementary lithography techniques include combinations of 193i, EUV, SADP, and E-beam. It is worth noting that these experiments are only examples to illustrate the usage of the framework. However, the output of the framework will strongly depend on the used parameters, thus the changing the parameters will lead to different conclusions about the technology.

Table 4 Number of vias in test cases.

Test case	Number of vias on V1	Number of vias on V3
AES	98,896	14,360
MIPS	86,939	7274
USB	99,366	7346

The framework is implemented in C++, using open access for layout manipulation. IBM CPLEX was used to solve the ILP. The experiments were run on a computing cluster, with a maximum of four threads on four cores and a total of 80G of virtual memory. The benchmarks were synthesized, placed, and routed using a projected 7-nm library from a leading IP provider, then the layouts were scaled down to 5-nm layouts. After scaling, the via dimension is 15 nm. All the experiments are performed on either the V1 layer or the V3 layer. The number of vias on these two layers in the used benchmarks is shown in Table 4.

The parameters used with different lithography techniques are shown in Table 5.

7.1 DSA+EUV SP Versus DSA+193i TP

In this study, we explore the feasibility of using EUV with one mask only to replace three masks in a 193i process to print the guiding templates for V1 layer. As shown in Table 5, a relatively big maximum group size was used (max_g = 7); while in 193i a smaller group size was used (max_g = 3) because the higher resolution of EUV can be used to achieve strongly confining templates having peanut shapes,[2] which result in less placement error.[23] For EUV, two scenarios are compared: one in which only manhattan DSA groups are allowed and one in which nonmanhattan groups are allowed as well.

Table 5 Parameters used in studies.

Parameter	Lithography	Value
min_dsa_pitch	All except SADP	27 nm
max_dsa_pitch	All except SADP	51 nm
min_dsa_pitch	SADP	48 nm
max_dsa_pitch	SADP	50 nm
max_g	193i	3
max_g	SADP	2
max_g	EUV	7
min_pitch_same_mask	193i	90 nm
min_pitch_same_mask	EUV	40 nm
min_pitch_diff_mask	193i	25 nm
min_pitch_diff_mask	EUV	22 nm

Table 6 Number of violations with SP EUV with manhattan DSA groups only, SP EUV with manhattan, and nonmanhattan DSA groups, 193i TP. Number of violations in case of 193i DP is also shown.

Testcase	DSA+EUV SP man.		DSA+EUV SP nonman		DSA+ 193i TP	DSA+ 193i DP
	Viol.	Runtime (min)	Viol.	Runtime (min)	Viol.	Viol.
AES	134	5.9	0	6.2	0	5930
MIPS	186	10.6	1	5.12	0	3476
USB	152	7.76	0	6.97	0	3977

The results of the experiment, in Table 6, show that EUV with nonmanhattan DSA groups can replace three masks of 193i since it has only one violation on one benchmark which occurred because of an off-grid via. However, EUV with manhattan groups only cannot. Our result for EUV with nonmanhattan groups agrees with the claim and empirical observation by Gronheid et al.[2] that DSA+EUV SP can be used to pattern via layer in 5-nm node.

7.2 DSA+193i TP+Unidirectional Templates Versus DSA+193i TP+Bidirectional Templates

As explained in Sec. 5.5, restricting the shapes on a mask to a certain direction can be beneficial to the process optimization. In this experiment, we evaluate the design-friendliness penalty of forcing all the groups on each mask to be unidirectional, using TP in which two masks are horizontal and one mask is vertical. The unidirectionality of the mask shapes did not sacrifice design-friendliness as shown in Table 7, which also shows the number of candidate DSA groups resulting from Sec. 4.2 and the number of selected DSA groups having more than one via in the design. Results show that the number of DSA groups has decreased, leading to more singletons (a guiding template printing one via only), which is expected since the undirectionality constraint has limited some groups. The number of candidate groups has not changed because the unidirectionality constraint has an effect only when the ILP is solved.

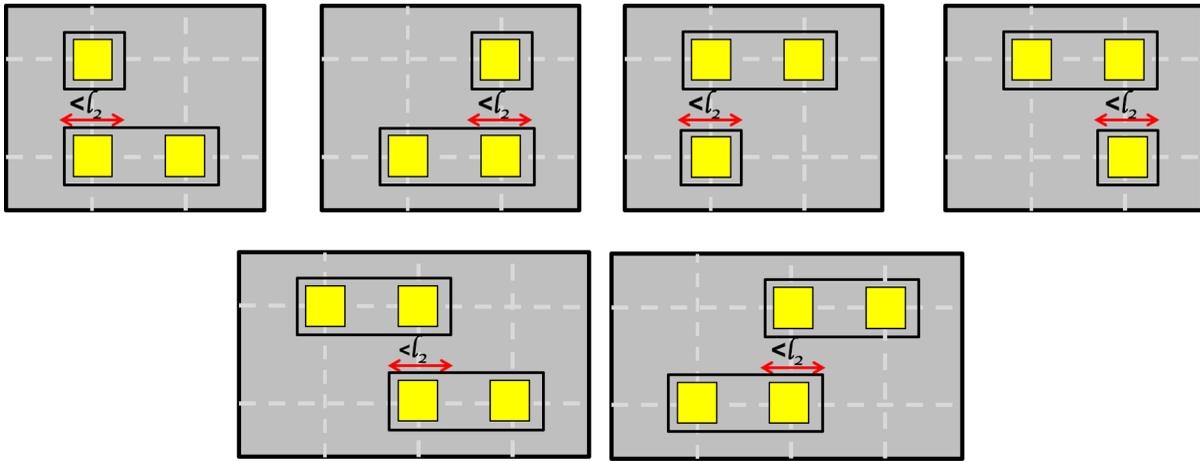**Table 8** DSA+193i+E-beam: percentage of shapes to print with E-beam with 193i and different number of masks.

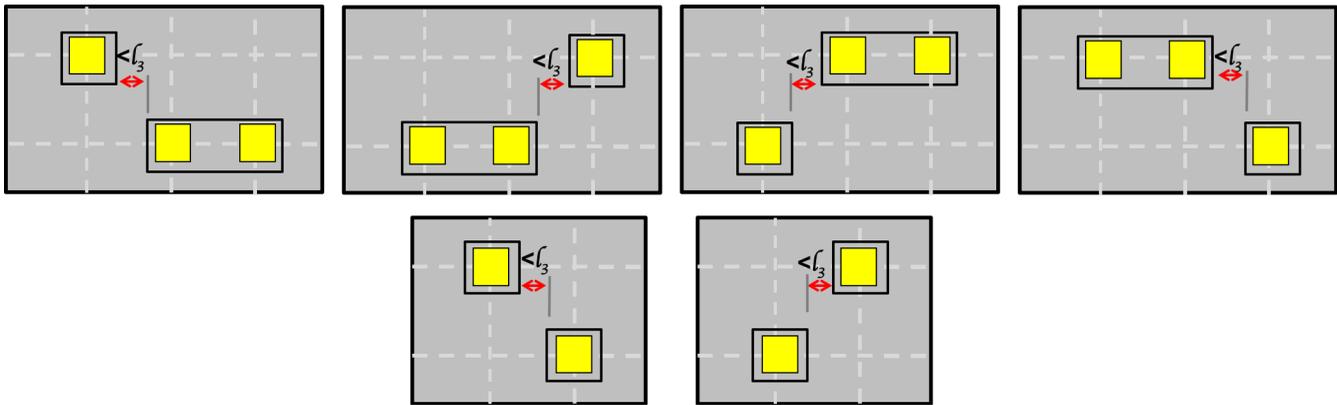| Testcase	DSA+193i SP+E-beam		DSA+193i DP+E-beam		DSA+193i TP+E-beam	
	% of E-beam	Runtime (s)	% of E-beam	Runtime (s)	% of E-beam	Runtime (s)
AES	92%	2.2	9%	2.5	0%	2.8
MIPS	88%	2.1	7%	2.5	0%	2.18
USB	89%	3.3	8%	3.2	0%	3.288

7.3 DSA+E-beam+193i

Hybrid lithography involving E-beam has already been studied in several works.[32,33] In this experiment, we consider a hybrid lithography process in which the guiding templates for DSA are printed using 193i lithography. Then, the templates that violate the min_pitch_same_mask are printed using E-beam, with the hope that the number of violations would be small enough such that the throughput is still not too low. The percentage of the vias that are in conflict and require their templates to be printed using E-beam is shown in Table 8. Assuming a threshold of 10%, it is clear that E-beam can likely save one mask exposure.

7.4 DSA+193i SADP

In this experiment, we study the feasibility of using SADP (using 193i) to pattern the templates for DSA. We use the SADP decomposition method used by Xu et al.,[34] in which tracks are alternated between mandrel and nonmandrel and the trim is used to create the vertical edges, which are the line ends. We use the SADP-friendly design rules used by Xu et al.,[34] which have been adapted from the work of Luk-Pat et al.[35] These design rules are translated into pattern-based rules (like hotspots). This experiment is run on the V3 layer for 7-nm layouts (without scaling to 5 nm). SADP is modeled as follows: the framework is run with one mask only. The first SADP rule (*OnTrackSpace*) has been enforced by setting min_pitch_same_mask to 59 nm, thus there is no need for pattern-based enforcing of the *OnTrackSpace* rule. However, the other three rules are enforced by representing

Table 7 Bidirectional DSA templates versus unidirectional templates on each mask (two horizontal masks and one vertical) versus on V1, using DSA+193i TP: number of violations, number of candidate groups, and number of selected groups.

Testcase	DSA+193i TP				DSA+193i TP			
	Bidirectional				Unidirectional			
	Viol.	Num cand. groups	Num groups	Runtime (min)	Viol.	Num cand. groups	Num groups	Runtime (min)
AES	0	54885	7432	2.8	0	54885	5819	2.9
MIPS	0	49587	5691	2.18	0	49587	4816	2.68
USB	0	56038	6666	3.3	0	56038	5618	3.4

Fig. 9 The forbidden patterns used to model SADP-friendly rule *OffTrackOverlap*, defined by Xu et al.[34] l_2 is the value of the *OffTrackOverlap* rule.

Fig. 10 The forbidden patterns used to model SADP-friendly rule *OffTrackSpace*, defined by Xu et al.[34] l_3 is the value of the *OffTrackSpace* rule.

the possible design rule violation as a forbidden pattern (hotspot), to be avoided. We used the following rule values:[34] $s\_r = 50$ nm, $w\_r = 50$ nm, $w\_e = 5$ nm, and $w\_sp = 40$ nm. No minimum area rule was enforced. The forbidden patterns used to model the *OffTrackOverlap* rule are shown in Fig. 9 and those used to model the *OffTrackSpace* rule are shown in Fig. 10. The patterns required to enforce the *OffTrackOffset* happen to be already included among the patterns of *OffTrackOverlap* rule. All the patterns are input to the framework in the format described by Badr et al.[27] and shown in Fig. 7.

Since SADP is more appropriate for regular layouts, we assume that the printed templates are all squares (for singletons) or rectangles (for groups of size two). Thus, we assume the templates do not have peanut shapes and this can increase the placement error of self-assembled holes.[23] To compensate, we allowed only groups of size two maximum, and we restricted the range of the self-assembly pitch to 2 nm: 48 to 50 nm. The guiding templates were designed as ellipses by Gharbi et al.[1] with an aspect ratio of $2:1$ to print groups of size two, and with an aspect ratio of $1:1$ to print singletons (dimension of square template for a singleton was assumed to be 40 nm). We adopt the same aspect ratio, but our templates are rectangles.

The results are shown in Table 9. We compare safe SADP, where trim edges can only print the vertical edges of the shapes and thus trim edges always lie in the middle of the sidewall; sensitive SADP, where trim edges are allowed to coincide with the spacer edge to have more relaxed design rules, eliminating the need for the *OffTrackSpace* rule; and SP. The overlay-sensitive SADP has a few violations, whereas safe SADP is not appropriate for patterning the templates in this scenario.

Table 9 Number of violations with overlay-safe SADP, overlay-sensitive SADP and SP on V3 layer.
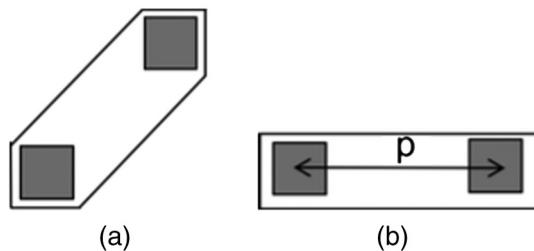
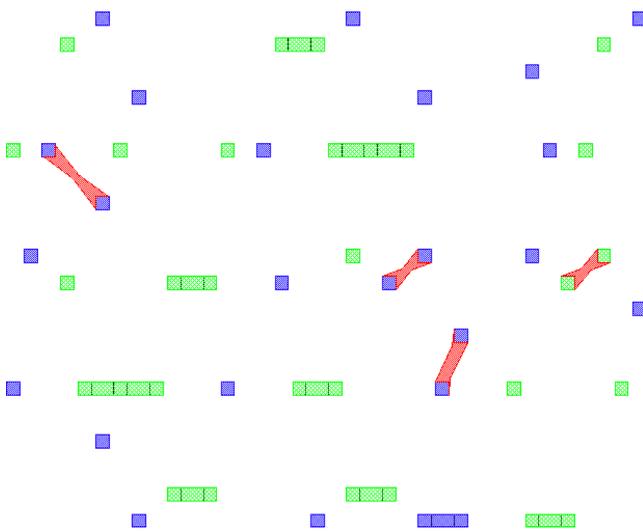| Testcase	DSA+SADP safe		DSA+SADP sensitive		DSA+193i SP	
	Viol.	Runtime (s)	Viol.	Runtime (s)	Viol.	Runtime (s)
AES	1169	12	17	28	1618	12
MIPS	342	13	9	22	468	9
USB	350	13	5	13	452	8

7.5 Minimum-Cost Technology Fix Experiments

As explained in Sec. 6, if the technology under evaluation is not friendly to the design, then the framework finds the minimum-cost change to the technology to remove all the violations. The computation of the technology change is run using one thread with maximum virtual memory of 16G. We do not use graph decomposition methods to solve it, because the technology change is global across the whole benchmark even if there are disconnected subgraphs of vias. The runtimes for the technology change finder ranges between 30 s to 4 h on our benchmarks. We used the default cost parameter values mentioned in Table 3.

7.5.1 DSA+193i DP

The use of DSA+193i DP was shown to be insufficient to pattern the guiding templates of V1 layer, as shown in Table 6. The benchmarks have pairs of vias that cannot be grouped because the two vias form an inclined DSA group [Fig. 11(a)] which is considered illegal by the used 193i grouping checker and the distance between the two vias is greater than the max_dsa_pitch [Fig. 11(b)]. In such cases, DSA was unable to resolve the violation through grouping. However, these were not the only reasons for failure; the minimum pitch is also too constrained for the dimensions and the configurations in the 5-nm

Fig. 11 Two reasons where DSA did not help remove the violations in DSA+193i DP. (a) Illegal grouping configuration. (b) Distance p < max dsapitch.

Fig. 12 Layout snippet for DSA+193i DP. Blue markers: DSA groups on mask 1, green markers: DSA groups on mask 2. Red markers: mask violations.

layouts leading to DP decomposition errors. A snippet of the result of decomposition and grouping for DSA+193i DP is shown in Fig. 12.

The computed technology fix is to change max_dsa_pitch to 60 nm (instead of 51 nm) and change min_pitch_same_mask to 78 nm (instead of 90 nm). The cost of the technology change is 41. Changing the cost parameters can result in different solutions.

7.5.2 DSA+SP EUV using manhattan DSA groups only

The results of using DSA+SP EUV with manhattan DSA groups only are shown in Table 6. The computed technology fix is to decrease the min_pitch_same_mask to 35 nm (instead of 40 nm), at a cost of 9. Using the alternative grouping checker that allows nonmanhattan groups would have removed the violations, but at a higher cost of 10.

7.5.3 DSA+SADP on V3

In the scenario of using DSA+safe SADP to print the V3 layer (Table 9), the suggested technology fix is to remove 9 out of 12 forbidden patterns representing the DSA rules. While for DSA+sensitive SADP, the suggested technology fix is to remove three out of six forbidden patterns. Interpreting this as a technology fix would mean violating SADP constraints. However, in this case, the technology fix is rather interpreted as removing the indicated patterns from the design, so the only way to be able to use DSA+SADP and benefit from the overlay advantages of SADP is to have a correct by construction DSA+SADP-aware design.

8 Conclusion

We have proposed a framework that can be used for path-finding for the hybrid DSA technologies in which a complementary lithography technique that is possibly multipatterned is used to print the guiding templates. Given the choice of the allowed groups, number of masks, design and mask rules, characteristics of BCP, and hotspots, the framework reports design-friendliness on the provided benchmarks. The framework is generic in the sense that it can be used to evaluate any type of hybrid DSA technology. Several case studies have been shown, including studies in which the complementary lithography technique is 193-nm immersion lithography, EUV, SADP, and E-beam.

References

1. A. Gharbi et al., "Contact holes patterning by directed self-assembly of block copolymers: what would be the Bossung plot?," *Proc. SPIE* **9049**, 90491N (2014).
2. R. Gronheid et al., "EUV patterned templates with grapho-epitaxy DSA at the N5/N7 logic nodes," *Proc. SPIE* **9776**, 97761W (2016).
3. J. Bekaert et al., "N7 logic via patterning using templated DSA: implementation aspects," *Proc. SPIE* **9658**, 965804 (2015).
4. Y. Ma et al., "Directed self assembly (DSA) compliant flow with immersion lithography: from material to design and patterning," *Proc. SPIE* **9777**, 97770N (2016).
5. Y. Ma et al., "Directed self-assembly graphoepitaxy template generation with immersion lithography," *J. Micro/Nanolith. MEMS MOEMS.* **14**(3), 031216 (2015).

6. Z. Xiao et al., "Contact pitch and location prediction for directed self-assembly template verification," in *20th Asia and South Pacific Design Automation Conf. (ASP-DAC '15)*, pp. 644–651, IEEE (2015).
7. G. Ouaknin et al., "Shape optimization for DSA," *Proc. SPIE* **9777**, 97770Y (2016).
8. Y. Du et al., "Block copolymer directed self-assembly (DSA) aware contact layer optimization for 10 nm 1D standard cell library," in *IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD '13)*, IEEE Press (2013).
9. H. Yi et al., "A general design strategy for block copolymer directed self-assembly patterning of integrated circuits contact holes using an alphabet approach," *Nano Lett.* **15**, 805–812 (2015).
10. Y. Du et al., "DSA-aware detailed routing for via layer optimization," *Proc. SPIE* **9049**, 90492J (2014).
11. S. Shim, W. Chung, and Y. Shin, "Defect probability of directed self-assembly lithography: fast identification and post-placement optimization," in *Proc. of the IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD '15)*, IEEE Press (2015).
12. S.-Y. Fang, Y.-X. Hong, and Y.-Z. Lu, "Simultaneous guiding template optimization and redundant via insertion for directed self-assembly," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 410–417, IEEE Press (2015).
13. J. Ou, B. Yu, and D. Z. Pan, "Concurrent guiding template assignment and redundant via insertion for DSA-MP hybrid lithography," in *Proc. of the 2016 on Int. Symp. on Physical Design (ISPD '16)*, pp. 39–46, ACM Press, New York (2016).
14. Z.-W. Lin and Y.-W. Chang, "Cut redistribution with directed self-assembly templates for advanced 1-D gridded layouts," in *21st Asia and South Pacific Design Automation Conf. (ASP-DAC '16)*, pp. 89–94, IEEE (2016).
15. Z.-W. Lin and Y.-W. Chang, "Double-patterning aware DSA template guided cut redistribution for advanced 1-D gridded designs," in *Proc. of the 2016 on Int. Symp. on Physical Design (ISPD '16)*, pp. 47–54, ACM Press, New York (2016).
16. W.-L. Wang et al., "A full-chip DSA correction framework," *Proc. SPIE* **9049**, 90491J (2014).
17. Y. Badr, J. A. Torres, and P. Gupta, "Incorporating DSA in multipatterning semiconductor manufacturing technologies," *Proc. SPIE* **9427**, 94270P (2015).
18. Y. Badr, A. Torres, and P. Gupta, "Mask assignment and synthesis of DSA-MP hybrid lithography for sub-7 nm contacts/vias," in *52nd ACM/EDAC/IEEE Design Automation Conf. (DAC '15)*, pp. 1–6 (2015).
19. J. Kuang and E. F. Young, "Simultaneous template optimization and mask assignment for DSA with multiple patterning," in *21st Asia and South Pacific Design Automation Conf. (ASP-DAC '16)*, pp. 75–82, IEEE (2016).
20. Z. Xiao et al., "Contact layer decomposition to enable DSA with multipatterning technique for standard cell based layout," in *21st Asia and South Pacific Design Automation Conf. (ASP-DAC '16)*, pp. 95–102, IEEE (2016).
21. I. Karageorgos et al., "Design strategy for integrating DSA via patterning in sub-7 nm interconnects," *Proc. SPIE* **9781**, 97810N (2016).
22. I. Karageorgos et al., "Design method and algorithms for directed self-assembly aware via layout decomposition in sub-7 nm circuits," *J. Micro/Nanolith. MEMS MOEMS* **15**(4), 043506 (2016).
23. Y. Ma et al., "Challenges and opportunities in applying grapho-epitaxy DSA lithography to metal cut and contact/via applications," *Proc. SPIE* **9231**, 92310T (2014).
24. Y. Badr and P. Gupta, "DSA_pathfind," http://nanocad.ee.ucla.edu/Main/DownloadForm (12 March 2017).
25. D. Ding, J. A. Torres, and D. Z. Pan, "High performance lithography hotspot detection with successively refined pattern identifications and machine learning," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **30**(11), 1621–1634 (2011).
26. V. Dai et al., "DRC plus: augmenting standard DRC with pattern matching on 2D geometries," *Proc. SPIE* **6521**, 65210A (2007).
27. Y. Badr, K.-W. Ma, and P. Gupta, "Layout pattern-driven design rule evaluation," *J. Micro/Nanolith. MEMS MOEMS.* **13**(4), 043018 (2014).
28. B. Yu et al., "Layout decomposition for triple patterning lithography," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2011).
29. T. Jhaveri et al., "Co-optimization of circuits, layout and lithography for predictive technology scaling beyond gratings," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **29**(4), 509–527 (2010).
30. J. Hopcroft and R. Tarjan, "Algorithm 447: efficient algorithms for graph manipulation," *Commun. ACM* **16**, 372–378 (1973).
31. J. W. Chinneck, *Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods*, Vol. **118**, Springer Science & Business Media, New York (2007).
32. Y. Yang et al., "Layout decomposition co-optimization for hybrid e-beam and multiple patterning lithography," in *The 20th Asia and South Pacific Design Automation Conf.*, pp. 652–657, IEEE (2015).
33. H. Tian et al., "Hybrid lithography for triple patterning decomposition and e-beam lithography," *Proc. SPIE* **9052**, 90520P (2014).
34. X. Xu et al., "Self-aligned double patterning aware pin access and standard cell layout co-optimization," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **34**(5), 699–712 (2015).
35. G. Luk-Pat et al., "Design compliance for spacer is dielectric (SID) patterning," *Proc. SPIE* **8326**, 83260D (2012).

Yasmine Badr received her BSc and MSc degrees in computer engineering from Cairo University. She is a PhD candidate in the Electrical Engineering Department at the University of California, Los Angeles (UCLA). Her research interests are in the computational methods for design and technology co-optimization.

Puneet Gupta received his PhD from the University of California, San Diego in 2007. He cofounded Blaze DFM Inc. (acquired by Tela Inc.) in 2004. He is a faculty member of the Electrical Engineering Department at UCLA. He has authored more than 130 papers, 16 U.S. patents, a book, and a book chapter. His research has focused on building high-value bridges across application-architecture-implementation-fabrication.