Technology Path-finding for Directed Self-Assembly for Via Layers

Yasmine Badr, Puneet Gupta

Electrical Engineering Department
University of California, Los Angeles
Email: ybadr@ucla.edu, puneet@ee.ucla.edu

ABSTRACT

Directed Self Assembly (DSA) is a very promising patterning technology for the sub-7nm 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. In this paper, 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 (DTCO). The framework optimally evaluates a DSA-based technology where an arbitrary lithography technique is used to print the guiding templates, possibly using many masks/exposures and provides a design-friendliness metric. The framework is used to evaluate technologies like DSA+193nm Immersion (193i) Lithography, DSA+Extreme Ultraviolet (EUV) and DSA+ 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.

1. INTRODUCTION

Directed Self Assembly (DSA) is a promising patterning technique for the sub-7nm 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 Figure 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 Figure 1.

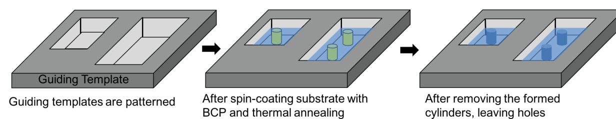


Figure 1: DSA process for contact/via holes.

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 193nm 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 Figure 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, while if 193i is used then only collinear groups are allowed, as shown in Figure 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). 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.

(a) Examples of DSA groups allowed in both 193i and EUV.

(b) Examples of DSA groups allowed in EUV but not in 193i.

Figure 2: Examples showing that the lithography technique used to print DSA guiding templates affects the allowed DSA groups

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. 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.

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.

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

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

The rest of the paper is organized as follows: Section 2 discusses the related work in 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 ILP formulation for path-finding is presented in Section 5. In Section 6, we show case studies that have been performed using the framework, followed by conclusion and future work in Section 7.

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 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 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 (SCFT) in the work of Ouaknin et al.[7] in addition to 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 of the

guiding templates. Yi et al.[9] show a design strategy (no automated design methods) for standard cell design based on the requirements of DSA technology, so it can not 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 et al.[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 which result in defective self-assembly through simulation.

The third category of research develops algorithms for hybrid technologies involving DSA, DSA+EUV and DSA+Multiple Patterning (MP) for 193i. Several works[17–20] perform DSA-aware mask assignment for DSA+193i technology. In addition, Ou et al.[13] solve the same problem while adding redundant vias, while Lin et al.[15] add cut redistribution. Karageorgos et al.[21] solve 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. Gronheid et al.[2] use 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 macro layout.

3. OVERVIEW OF THE FRAMEWORK

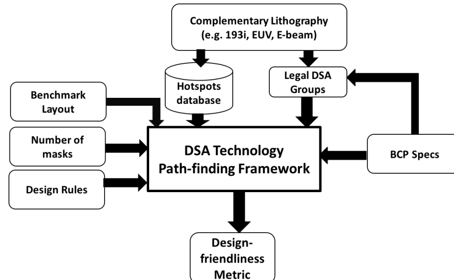


Figure 3: Overview of the Hybrid DSA Technology Exploration Framework

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

BCP Specifications. These are the natural pitch L_0 of the BCP (which is also assumed to be the minimum realizable pitch by the BCP), the maximum pitch to which the BCP can be stretched (max_dsa_pitch), assembled holes dimension and the maximum allowed number of vias per DSA group (max_g). The max_g constraint exists because earlier research has shown that smaller DSA group sizes can lead to more robust self-assembly.[22]

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

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 *. 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

*The $min_pitch_diff_mask$ rule should be satisfied even if the shapes are assigned to different masks because of overlay error.

details are provided in Section 4.5.

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

Table 1: Definition of Input Parameters

Parameter	**Definition**
L_0	natural pitch of BCP
max_dsa_pitch	maximum pitch to which the BCP can be stretched
max_g	maximum allowed number of vias per DSA group
$min_pitch_same_mask$	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

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 model any DSA-based technology. The reasons for this generalization capability are: first, the foundry can define and use its own grouping checker with custom/non-conventional 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 DSA PATH-FINDING FRAMEWORK

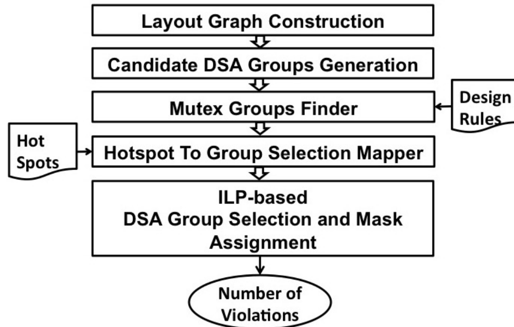


Figure 4: Flow of the DSA Path-finding Framework

The flow of the proposed framework is shown in Figure 4. First, the candidate DSA groups are generated. Then the pairs of groups which can not co-exist are determined. After that, the group combinations which will result in a forbidden pattern if assigned to the same mask are found. Finally, the output of the previous steps is used to formulate and solve an Integer Linear Program (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 $min_pitch_same_mask$.

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 single-connected-component subgraphs, that contain this node and with 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. Although the

number of single-component subgraphs is exponential, practically the number of subgraphs is bounded due to the constraint that for each node, we only enumerate the subgraphs having number of nodes less than or equal to max_g.

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 non-compliant ones. Grouping checkers are explained in Section 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 like the collinear grouping checker typically used for 193i and the more flexible grouping checker which is used for EUV experiments. Other options are also provided like 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.

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[†], and the center-to-center distance between every two neighboring vias in the group is within the allowed BCP pitch range.

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

1. Groups whose graphs, similar to the graph explained in Section 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 non-manhattan group which can be determined legal or illegal according to the the used knob allowing or disallowing non-manhattan neighborhood.

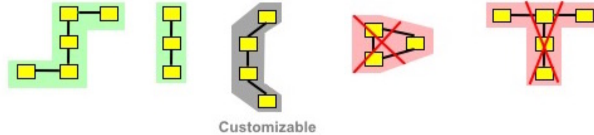


Figure 5: Examples of Legal and Illegal groups according to our EUV grouping checker. A line between two vias means that distance between their centers is less than $min_pitch_same_mask$

4.4 Mutually Exclusive (Mutex) Groups Finder

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

MUTEX Case 1. A set of groups have one (or more) common via(s). Out of all the candidate groups involving a certain via, only one group can be selected.

[†] All vias must be square shapes with a dimension equal to the assembled hole diameter.

MUTEX Case 2. Two groups have a pitch smaller than $min_pitch_diff_mask$. Thus only one of these groups can be selected (See the definition of the used design rules in Section 3). Note that if a singleton (i.e. non-grouped via) is mutually exclusive with a non-singleton group because of this reason, the non-singleton group is removed from the grouping options. This is because the non-singleton group will result in a design rule violation, regardless 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[‡]. Thus the input knobs of the framework can disable this case.
MUTEX Case 4. Two groups have a pitch smaller than $min_pitch_same_mask$. The two groups can be selected only if they are assigned to different masks.
To find the mutex groups belonging to cases 2, 3 and 4 described above, the neighborhood of each via is examined in order to find such pairs of groups. Mutex groups are fed into the ILP formulation.

4.5 Hot Spot to Group Selection Mapper

In addition to the mutex groups described in Section 4.4, some groups can not 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.[23]
2. Complex design rule. In advanced nodes, foundries had to introduce a lot of complex 2D and conditional rules. These rules can require pattern-based representation.[24]
3. Forbidden pattern due to using a restrictive patterning technology like Self-Aligned Multiple Patterning.

Thus in order to have a correct evaluation for the technology, the generated DSA templates must be hotspot-free. Moreover, by 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.,[25] where the *segment* representation is used to encode 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, Figure 6 shows an example of a 2x2 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.

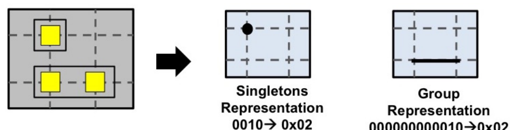


Figure 6: A hotspot and its corresponding representation

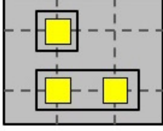
A hotspot on a mask is defined by a list of *segments* that are occupied by DSA groups, a list of *segments* that are empty, a list of *nodes* that are filled due to singletons, and a list of *nodes* that are empty (i.e. no singletons exist at the *node* location).

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 non-empty windows, and generating the forbidden combinations of groups. For each hotspot and for each non-empty 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 which 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 which 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

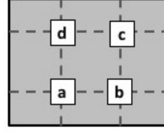
[‡]This can be done if self-assembly is done for each mask then the assembled holes are transferred to a hard mask

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 Figure 7a to exist in the layout window shown in Figure 7b, 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}.



(a) Hotspot



(b) Layout Window

Figure 7: Example showing the different sets of groups generated for one hypothetical hotspot and one 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. §

The used constraints are derived from the input to the framework described in Section 3, the candidate DSA groups as explained in Section 4.2, the mutex groups as described in Section 4.4 as well as the forbidden group combinations due to hotspots as explained in Section 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 1 mask (Single Patterning (SP)), 2 masks (Double Patterning (DP)), 3 masks (Triple Patterning (TP)) or 4 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.¶

5.1 Objective Function

The objective function in Equation (1) aims at minimizing the number of conflicts. As explained earlier, a conflict exists between two vias if there is graph edge between them and they are not in the same DSA group.

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

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 Equation (2). The constraints in Equations (3) 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

§ In case of evaluating a single exposure technology, same formulation is used but there will be no mask or similarity variables, so the result is only the group selection.

¶In case of TP, other constraints are added in order to prohibit the unused mask bit combinations, similar to the work of Yu et al. [26] and Badr et al. [18]

Table 2: Notation used in ILP Formulation

Notation	Description
c_{ij}	Variable indicating if i^{th} and j^{th} vias are in conflict
m_i^b	b^{th} bit of mask index of ith via
s_{ij}^b	Similarity variable indicating if b^{th} bits in masks of ith and j^{th} vias are identical
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 **I** are grouped. Variable only exists if the vias in set **I** form a candidate group and if $\lvert\mathbf{I}\rvert \geq 2$
$dir(\mathbf{P_k})$	Orientation of the candidate DSA group formed of $\mathbf{P_k}$. Value is 'v' if vertical; 'h' if horizontal; 'o' if non-collinear.
$\mathbf{E_m}$	m^{th} set of mutex DSA groups
M	Number of sets of mutex DSA groups of MUTEX Cases 1-3
N	Number of sets of mutex DSA groups of MUTEX Case 4
Notation for the Hot Spot Prevention Constraints	
$\mathbf{ONG}_{wq}^h$	q^{th} Set of ON Groups for the hth hotspot, for the wth layout window
$\mathbf{OFFG}_w^h$	Set of OFF Groups for the hth hotspot, for the wth layout window
$\mathbf{AS}_w^h$	Set of Absent Singletons for the hth hotspot, for the wth layout window
n_{wq}^h	Index of an arbitrary via in an arbitrary group in $\mathbf{ONG}_{wq}^h$
f_w^h	Index of an arbitrary via in an arbitrary group in $\mathbf{OFFG}_w^h$
$\mathbf{N}_{wq}^{hy}$	y^{th} group in $\mathbf{ONG}_{wq}^h$
$\mathbf{F}_w^{hy}$	y^{th} group in $\mathbf{OFFG}_w^h$

the two corresponding mask bits. In addition, the constraints in Equations (4) are added in order to only allow the selection of the DSA group if all the involved vias are assigned to the same mask.

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

$$s_{ij}^1 \geq 1 - m_i^1 - m_j^1 \qquad \forall\, (i,j) \in \mathbf{GEs} \tag{3a}$$
$$s_{ij}^1 \leq 1 - m_i^1 + m_j^1 \qquad \forall\, (i,j) \in \mathbf{GEs} \tag{3b}$$
$$s_{ij}^1 \leq 1 + m_i^1 - m_j^1 \qquad \forall\, (i,j) \in \mathbf{GEs} \tag{3c}$$
$$s_{ij}^1 \geq -1 + m_i^1 + m_j^1 \qquad \forall\, (i,j) \in \mathbf{GEs} \tag{3d}$$
$$s_{ij}^2 \geq 1 - m_i^2 - m_j^2 \qquad \forall\, (i,j) \in \mathbf{GEs} \tag{3e}$$
$$s_{ij}^2 \leq 1 - m_i^2 + m_j^2 \qquad \forall\, (i,j) \in \mathbf{GEs} \tag{3f}$$
$$s_{ij}^2 \leq 1 + m_i^2 - m_j^2 \qquad \forall\, (i,j) \in \mathbf{GEs} \tag{3g}$$
$$s_{ij}^2 \geq -1 + m_i^2 + m_j^2 \qquad \forall\, (i,j) \in \mathbf{GEs} \tag{3h}$$

$$s_{ij}^1 \geq g_{\mathbf{P}_k} \; \forall \{i,j,k | (i,j) \in \mathbf{GEs}, \{i,j\} \subseteq \mathbf{P}_k, k \in [1..K]\} \tag{4a}$$
$$s_{ij}^2 \geq g_{\mathbf{P}_k} \; \forall \{i,j,k | (i,j) \in \mathbf{GEs}, \{i,j\} \subseteq \mathbf{P}_k, k \in [1..K]\} \tag{4b}$$

5.3 Mutual Exclusive Group Constraints

As explained in Section 4.4, some groups can not co-exist due to MUTEX cases 1-4.

Constraints for MUTEX Cases 1-3

For MUTEX cases 1-3, constraints in Equation (5) are generated to prohibit the selection of more than one group from each set of MUTEX groups.

$$\sum_{g \in \mathbf{E_i}} g \leq 1 \quad \forall\, i \in [1..M] \tag{5}$$

Constraints for MUTEX Case 4

For MUTEX case 4, two mutually exclusive groups can co-exist only if they are assigned to different masks. The constraints in Equations (6) and (7) 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 non-singletons and in the case of one of the two groups being a singleton, respectively.

$$\begin{aligned} & g_A + g_B + s^1_{xy} + s^2_{xy} \leq 3 \quad \forall n \in [1..N] \\ & s.t. \quad \mathbf{E}_n = \{\mathbf{A}, \mathbf{B}\}, |\mathbf{A}| \geq 2, |\mathbf{B}| \geq 2, x \in \mathbf{A}, y \in \mathbf{B} \end{aligned} \tag{6}$$

$$\begin{aligned} & g_{\mathbf{A}} + s^1_{xy} + s^2_{xy} \leq 2 \quad \forall n \in [1..N] \\ & s.t. \quad \mathbf{E}_n = \{\mathbf{A}, \mathbf{B}\}, |\mathbf{A}| \geq 2, x \in \mathbf{A}, \mathbf{B} = \{y\} \end{aligned} \tag{7}$$

5.4 Hotspot Prevention Constraints

Constraints are added in order to prevent the existence of guiding templates which create hotspots. As explained in Section 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 Equations (8) 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.

$$\begin{aligned} & \sum_{\mathbf{A} \in \mathbf{ONG}^h_{wq}} g_{\mathbf{A}} + \sum_{\mathbf{B} \in \mathbf{OFFG}^h_w} (1 - g_{\mathbf{B}}) + \sum_{k=1}^{k=2} \sum_{\substack{x \in \mathbf{AS}^h_w \\ x \neq (n^h_{wq})}} (1 - s^k_{x(n^h_{wq})}) \\ & + \sum_{k=1}^{k=2} \sum_{\substack{y=1 \\ i \in \mathbf{N}^{hy}_{wq} \\ j \in \mathbf{N}^{h(y+1)}_{wq} \\ i \neq j}}^{y=\left|\mathbf{ONG}^h_{wq}\right|-1} s^k_{ij} + \sum_{k=1}^{k=2} \sum_{\substack{z=1 \\ i \in \mathbf{F}^{hz}_w \\ j \in \mathbf{F}^{h(z+1)}_w \\ i \neq j}}^{z=\left|\mathbf{OFFG}^h_w\right|-1} s^k_{ij} + \sum_{k=1}^{k=2} s_{(n^h_{wq})(f^h_w)} \\ & \leq 3(\left|\mathbf{ONG}^h_{wq}\right| + \left|\mathbf{OFFG}^h_w\right|) + 2\left|\mathbf{AS}^h_w\right| - 3 \quad \forall\, q, h, w \end{aligned} \tag{8}$$

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.[27] 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 Equations (9) 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 which are aligned on the same Y-axis whereas a horizontal group is a group of two or more vias which 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.[1]

$$g_{\mathbf{P}_k} + m^1_i \leq 1 \quad \forall \{k, i | k \in [1..K], i \in \mathbf{P}_k, dir(\mathbf{P}_k) = `v'\} \tag{9a}$$

$$g_{\mathbf{P}_k} - m^1_i \leq 0 \quad \forall \{k, i | k \in [1..K], i \in \mathbf{P}_k, dir(\mathbf{P}_k) = `h'\} \tag{9b}$$

5.6 Parallelization

It is required to solve the ILP in parallel in order to reduce runtime without sacrificing optimality. Thus, the **connected components**[28] 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. CASE STUDIES AND RESULTS

In this section, we present several exploration studies which 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.

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 4 threads on four cores and total of 80G of virtual memory. The benchmarks were synthesized, placed and routed using a projected 7nm library from a leading IP provider, then the layouts were scaled down to 5nm layouts. After scaling, the via dimension is 15nm. All the experiments are either performed on the V1 layer or the V3 layer. The number of vias on these two layers in the used benchmarks is shown in Table 3.

The parameters used with different lithography techniques are shown in Table 4.

Table 3: Number of vias in test cases

Test case	Number of Vias on V1	Number of Vias on V3
AES	98896	14360
MIPS	86939	7274
USB	99366	7346

Table 4: Parameters used in studies

Parameter	**Lithography**	**Value**
L_0	all except SADP	27nm[1]
max_dsa_pitch	all except SADP	51nm
L_0	SADP	48nm
max_dsa_pitch	SADP	50nm
max_g	193i	3
max_g	SADP	2
max_g	EUV	7
$min_pitch_same_mask$	193i	90nm
$min_pitch_same_mask$	EUV	40nm
$min_pitch_diff_mask$	193i	25nm
$min_pitch_diff_mask$	EUV	22nm

6.1 DSA+ EUV SP vs. DSA+193i TP

In this study, we explore the feasibility of using EUV with 1 mask only to replace three masks in a 193i process to print the guiding templates for V1 layer. As shown in Table 4, 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.[22] For EUV, two scenarios are compared: one where only manhattan DSA groups are allowed and one where non-manhattan groups are allowed as well.

The results of the experiment, in Table 5, show that EUV with non-manhattan groups can replace three masks of 193i since it has only one violation on one benchmark which occured because of an off-grid via. However, EUV with manhattan groups only can not. Our result for EUV with non-manhattan 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 5nm node.

6.2 DSA+ 193i TP + Unidirectional templates vs. DSA + 193i TP + Bidirectional templates

As explained in Section 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 the all the groups on each mask to be unidirectional, using TP where two masks are horizontal and one mask is vertical. The uni-directionality of the mask shapes did not sacrifice design-friendliness as shown in Table 6, which also shows the number of candidate DSA groups resulting from Section 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

Table 5: Number of violations with SP EUV with manhattan DSA groups only, SP EUV with manhattan and non-manhattan groups, 193i TP. Number of Violations in case of 193i DP is also shown.

Testcase	DSA+EUV SP man.		DSA+EUV SP non-man		DSA +193i TP	DSA+193i DP
	Viol.	Runtime(m)	Viol.	Runtime(m)	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

Table 6: Bidirectional DSA templates vs. Unidirectional templates on each mask (2 horizontal masks and 1 vertical) vs. on V1, using DSA+193i TP: Number of violations, Number of candidate groups and number of selected groups

Testcase	DSA+193i TP Bidirectional				DSA+193i TP Unidirectional			
	Viol.	Num Cand. Groups	Num Groups	Runtime(m)	Viol.	Num Cand. Groups	Num Groups	Runtime(m)
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

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.

6.3 DSA+ E-beam + 193i

Hybrid lithography involving E-beam has already been studied in several works.[29,30] In this experiment we consider a hybrid lithography process where the guiding templates for DSA are printed using 193i lithography. Then the templates which 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 which are in conflict and require their templates to be printed using e-beam is shown in Table 7. It is clear that E-beam can likely save one mask exposure.

6.4 DSA+ SADP

In this experiment, we study the feasibility of using SADP to pattern the templates for DSA. We use the SADP decomposition method from,[31] where tracks are alternated between between mandrel and non-mandrel 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.,[31] which have been adapted from the work of Luk-Pat et al.[32] These design rules are translated into pattern-based rules (like hotspots). Examples are shown in Figure 8.This experiment is run on the V3 layer for 7nm layouts (without scaling to 5nm). SADP is modeled as follows: the framework is run with one mask only. The first SADP rule (*OnTrackSpace*) is enforced by setting *min_pitch_same_mask* to 59nm, thus there is no need for pattern-based enforcing of the *OnTrackSpace* rule. However the other three rules are enforced by representing the possible design rule violation as a hotspot, to be avoided. We used the following rule values:[31] s_r=50nm, w_r=50nm, w_e=5nm, w_sp=40nm. No minimum area rule was enforced.

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

Table 7: 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 SP +E-beam		DSA +193i TP +E-beam	
	% of Ebeam	Runtime(s)	% of Ebeam	Runtime(s)	% of Ebeam	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

and this can increase the placement error of self-assembled holes.[22] To compensate, we only allowed groups of size two maximum, and we restricted the range of the self-assembly pitch to 2 nm: 48nm-50nm. 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 40nm). We adopt the same aspect ratio, but our templates are rectangles.

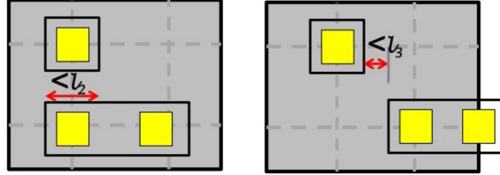


Figure 8: Samples of the forbidden patterns used to model SADP-friendly rules *OffTrackOverlap* and *OffTrackSpace*, defined by Xu et al.[31] l_2 and l_3 are the values of the *OffTrackOverlap* and *OffTrackSpace* rules respectively.

The results are shown in Table 8. 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 Off-track Space rule; and Single Patterning (SP). The overlay-sensitive SADP has a few violations, while safe SADP is not appropriate for patterning the templates in this scenario.

Table 8: 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. 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 multi-patterned is used to print the guiding templates. Given, the choice of the allowed groups, number of masks, design and mask rules, characteristics of block copolymer 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 where the complementary lithography technique is 193nm immersion lithography, EUV, SADP and E-beam.

REFERENCES

[1] Gharbi, A., Tiron, R., Argoud, M., Chevalier, X., Belledent, J., Pradelles, J., Barros, P. P., Navarro, C., Nicolet, C., Hadziioannou, G., et al., "Contact holes patterning by directed self-assembly of block copolymers: What would be the bossung plot?," in [*SPIE Advanced Lithography*], 90491N–90491N (2014).

[2] Gronheid, R., Boeckx, C., Doise, J., Bekaert, J., Karageorgos, I., Ruckaert, J., Chan, B. T., Lin, C., and Zou, Y., "Euv patterned templates with grapho-epitaxy dsa at the n5/n7 logic nodes," in [*SPIE Advanced Lithography*], 97761W–97761W, SPIE (2016).

[3] Bekaert, J., Doise, J., Gronheid, R., Ryckaert, J., Vandenberghe, G., Fenger, G., Her, Y. J., and Cao, Y., "N7 logic via patterning using templated DSA: implementation aspects," in [*Photomask Japan 2015*], Yoshioka, N., ed., 965804, SPIE (jul 2015).

[4] Ma, Y., Wang, Y., Word, J., Lei, J., Mitra, J., Torres, J. A., Hong, L., Fenger, G., Khaira, D., Preil, M., Yuan, L., Kye, J., and Levinson, H. J., "Directed Self Assembly (DSA) compliant flow with immersion lithography: from material to design and patterning," in [*SPIE Advanced Lithography*], 97770N, SPIE (mar 2016).

[5] Ma, Y., Lei, J., Torres, J. A., Hong, L., Word, J., Fenger, G., Tritchkov, A., Lippincott, G., Gupta, R., Lafferty, N., et al., "Directed self-assembly graphoepitaxy template generation with immersion lithography," *JM3* **14**(3), 031216–031216 (2015).

[6] Xiao, Z., Du, Y., Wong, M. D., Yi, H., Wong, H., and Zhang, H., "Contact pitch and location prediction for directed self-assembly template verification," in [*ASPDAC*], 644–651, IEEE (2015).

[7] Ouaknin, G., Laachi, N., Delaney, K., Fredrickson, G., and Gibou, F., "Shape optimization for DSA," in [*SPIE Advanced Lithography*], Bencher, C. and Cheng, J. Y., eds., 97770Y, SPIE (mar 2016).

[8] Du, Y., Guo, D., Wong, M. D. F., Yi, H., Wong, H.-S. P., Zhang, H., and Ma, Q., "Block copolymer directed self-assembly (DSA) aware contact layer optimization for 10 nm 1d standard cell library," in [*Proc. ICCAD*], *ICCAD '13*, IEEE Press (2013).

[9] Yi, H., Bao, X.-Y., Tiberio, R., and Wong, H.-S. P., "A general design strategy for block copolymer directed self-assembly patterning of integrated circuits contact holes using an alphabet approach.," *Nano letters* **15**, 805–12 (feb 2015).

[10] Du, Y., Xiao, Z., Wong, M. D., Yi, H., and Wong, H.-S. P., "Dsa-aware detailed routing for via layer optimization," in [*SPIE Advanced Lithography*], 90492J–90492J (2014).

[11] Shim, S., Chung, W., and Shin, Y., "Defect probability of directed self-assembly lithography: Fast identification and post-placement optimization," in [*ICCAD*], IEEE Press (2015).

[12] Fang, S.-Y., Hong, Y.-X., and Lu, Y.-Z., "Simultaneous guiding template optimization and redundant via insertion for directed self-assembly," in [*ICCAD*], 410–417, IEEE Press (2015).

[13] Ou, J., Yu, B., and Pan, D. Z., "Concurrent Guiding Template Assignment and Redundant via Insertion for DSA-MP Hybrid Lithography," in [*Proceedings of the 2016 on International Symposium on Physical Design - ISPD '16*], 39–46, ACM Press, New York, New York, USA (apr 2016).

[14] Lin, Z.-W. and Chang, Y.-W., "Cut redistribution with directed self-assembly templates for advanced 1-D gridded layouts," in [*2016 21st ASP-DAC*], 89–94, IEEE (jan 2016).

[15] Lin, Z.-W. and Chang, Y.-W., "Double-Patterning Aware DSA Template Guided Cut Redistribution for Advanced 1-D Gridded Designs," in [*Proceedings of ISPD '16*], 47–54, ACM Press, New York, New York, USA (apr 2016).

[16] Wang, W.-L., Latypov, A., Zou, Y., and Coskun, T., "A full-chip dsa correction framework," in [*SPIE Advanced Lithography*], 90491J–90491J, SPIE (2014).

[17] Badr, Y., Torres, J. A., and Gupta, P., "Incorporating dsa in multipatterning semiconductor manufacturing technologies," in [*SPIE Advanced Lithography*], (2015).

[18] Badr, Y., Torres, A., and Gupta, P., "Mask assignment and synthesis of dsa-mp hybrid lithography for sub-7nm contacts/vias," in [*DAC*], 1–6 (June 2015).

[19] Kuang, J. and Young, E. F., "Simultaneous template optimization and mask assignment for DSA with multiple patterning," in [*2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*], 75–82, IEEE (jan 2016).

[20] Xiao, Z., Lin, C.-X., Wong, M. D., and Zhang, H., "Contact layer decomposition to enable DSA with multi-patterning technique for standard cell based layout," in [*2016 ASPDAC*], 95–102, IEEE (jan 2016).

[21] Karageorgos, I., Ryckaert, J., Tung, M. C., Wong, H. S. P., Gronheid, R., Bekaert, J., Karageorgos, E., Croes, K., Vandenberghe, G., Stucchi, M., and Dehaene, W., "Design strategy for integrating DSA via patterning in sub-7 nm interconnects," in [*SPIE Advanced Lithography*], Capodieci, L. and Cain, J. P., eds., 97810N, SPIE (mar 2016).

[22] Ma, Y., Torres, J. A., Fenger, G., Granik, Y., Ryckaert, J., Vanderberghe, G., Bekaert, J., and Word, J., "Challenges and opportunities in applying grapho-epitaxy dsa lithography to metal cut and contact/via applications," in [*EMLC*], (2014).

[23] Ding, D., Torres, J. A., and Pan, D. Z., "High performance lithography hotspot detection with successively refined pattern identifications and machine learning," *TCAD* (2011).

[24] Dai, V., Yang, J., Rodriguez, N., and Capodieci, L., "Drc plus: augmenting standard drc with pattern matching on 2d geometries," in [*SPIE Advanced Lithography*], (2007).

[25] Badr, Y., Ma, K.-w., and Gupta, P., "Layout pattern-driven design rule evaluation," *JM3* **13**(4), 043018 (2014).

[26] Yu, B., Yuan, K., Zhang, B., Ding, D., and Pan, D., "Layout decomposition for triple patterning lithography," in [*Proc. ICCAD*], (2011).

[27] Jhaveri, T., Rovner, V., Liebmann, L., Pileggi, L., Strojwas, A. J., and Hibbeler, J. D., "Co-optimization of circuits, layout and lithography for predictive technology scaling beyond gratings," *TCAD* **29**(4), 509–527 (2010).

[28] Hopcroft, J. and Tarjan, R., "Algorithm 447: Efficient algorithms for graph manipulation," *Commun. ACM* **16**, 372–378 (June 1973).

[29] Yang, Y., Luk, W.-S., Zhou, H., Yan, C., Zeng, X., and Zhou, D., "Layout decomposition co-optimization for hybrid e-beam and multiple patterning lithography," in [*The 20th Asia and South Pacific Design Automation Conference*], 652–657, IEEE (2015).

[30] Tian, H., Zhang, H., Xiao, Z., and Wong, M. D., "Hybrid lithography for triple patterning decomposition and e-beam lithography," in [*SPIE Advanced Lithography*], 90520P–90520P, International Society for Optics and Photonics (2014).

[31] Xu, X., Cline, B., Yeric, G., Yu, B., and Pan, D. Z., "Self-aligned double patterning aware pin access and standard cell layout co-optimization," *TCAD* **34**(5), 699–712 (2015).

[32] Luk-Pat, G., Miloslavsky, A., Painter, B., Lin, L., Bisschop, P., and Lucas, K., "Design compliance for spacer is dielectric (sid) patterning," in [*Proc. SPIE*], **8326**, 83260D (2012).