Self-aligned Double Patterning Layout Decomposition with Complementary E-Beam Lithography

Jhih-Rong Gao, Bei Yu, and David Z. Pan
Dept. of ECE, The University of Texas at Austin, Austin, TX 78712
Email: {jrgao, bei, dpan}@cerc.utexas.edu

Abstract— Advanced lithography techniques enable higher pattern resolution; however, techniques such as extreme ultraviolet lithography and e-beam lithography (EBL) are not yet ready for high volume production. Recently, complementary lithography has become promising, which allows two different lithography processes work together to achieve high quality layout patterns while not increasing much manufacturing cost. In this paper, we present a new layout decomposition framework for self-aligned double patterning and complementary EBL, which considers overlay minimization and EBL throughput optimization simultaneously. We perform conflict elimination by merge-and-cut technique and formulate it as a matching-based problem. The results show that our approach is fast and effective, where all conflicts are solved with minimal overlay error and e-beam utilization.

I. Introduction

The semiconductor industry demands advanced lithography to enable small feature size and high density designs. Currently, 193nm immersion lithography (193i) is still the mainstream for 32nm and 22nm technology nodes. Incorporating double patterning lithography (DPL) and multiple patterning lithography (MPL) can keep pushing the resolution limit [1]. However, DPL/MPL requires strict layout compliance, and to fabricate a layout with multiple masks increases the design complexity. Although some leading lithography technologies, such as extreme ultraviolet lithography (EUVL) [2], e-beam lithography (EBL) [3], direct self-assembly [4], etc., achieve good pattern quality, they are not yet ready for high volume production.

Complementary lithography is proposed to allow 193nm optical lithography work hand-in-hand with high-resolution lithography to enable advanced designs [5]. In the first step, base features are created by cheaper optical lithography or self-aligned double patterning (SADP); in the second step, high-resolution lithography techniques are applied to cut unnecessary lines. Such line cutting can be accomplished by costly quadruple patterning, EUVL, or EBL. By carefully arranging how features are generated with the combined lithography techniques, we can achieve good pattern quality with a reasonable manufacturing cost. The advantages of adopting complementary lithography include: (1) high throughput by generating base patterns with mature optical lithography; and (2) improved mask yield by partial EUVL or EBL patterning, while no heavy manufacturing cost introduced.

Recently, the technique to combine optical and complementary EBL becomes promising. Lam et al. [6, 7] proposed us-

ing EBL to complement 193nm immersion lithography for 1D layout. The choice of SADP enables better overlay control compared with conventional Litho-Etch-Litho-Etch type double patterning. EBL is a maskless lithography which directly writes high-resolution patterns into the silicon wafer with e-beams. Although having their own advantage, standalone SADP and EBL are limited by low manufacturing flexibility and low throughput, respectively. By combining SADP with EBL together, we can improve manufacturability because EBL provides higher resolution; in the mean time, we need to reduce EBL utilization to ensure high productivity. This requires an overall optimization to consider complementary technique in the layout decomposition flow.

Several studies [6–8] have presented the effectiveness of applying SADP with line cutting technique on 1D layout designs. In order to optimize the overall throughput with hybrid SADP and EBL, an integer linear programming (ILP) -based approach [9] was proposed by properly distributing cutting patterns to the optical mask and e-beams. However, this work only targets at 1D gridded designs and allows wire-end extension that is not always permitted for general designs. There have been some studies [10–13] presented for pure SADP layout decomposition of 2D random patterns, where layout decomposition is the process to assign layout features into two different fabrication steps. These approaches impose strict SADP process rules to ensure the decomposed layout is SADP-manufacturable. However, the design flexibility is restricted and the layout may still not be decomposable given a complex layout.

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

- We present a new layout decomposition framework for SADP and complementary EBL, which considers overlay minimization and EBL throughput optimization simultaneously.

- We propose a new graph formulation and a matching-based approach that allows eliminating conflicts by the merge-and-cut technique.

- We show that for pure SADP layout decomposition problem, our approach can be adapted to minimize conflicts with overlay consideration.

- The results show that our approach is very efficient, and that all conflicts can be eliminated with minimal overlay error and e-beam utilization.

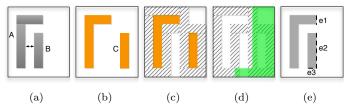In the rest of this paper, we will introduce SADP process

Fig. 1. SADP process. (a) Target layout patterns. (b) The mandrel mask. (c) Spacer deposition. (d) Mandrel removal and the trim mask. (e) Final patterns.

Fig. 2. Merge-and-cut example. (a) Target layout. (b) Two-coloring result. (c)~(e) Layout decomposition with merge-and-cut. Red lines show boundaries with overlay error risk.

and the layout decomposition problem in Sec. II. In Sec. III, we present our *face graph* formulation that embeds SADP constraints as well as the solution candidates for solving conflicts. Our layout decomposition that performs simultaneous overlay and EBL throughput optimization is presented in Sec. IV. We then explain an adapted conflict minimization approach that can be used for pure SADP in Sec. V. Finally, we will show experimental results in Sec. VI, followed by the conclusion in Sec. VII.

II. SADP AND CONFLICT SOLVING TECHNIQUE

A. SADP Process Overview

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

Fig. 1 shows the SADP process, where the arrow indicates a conflict between the two target patterns, meaning they cannot be fabricated on the same mask. Part of the target layout is first defined by the mandrel mask as shown in Fig. 1(b). Pattern C is called an *assist mandrel*, which helps to define target patterns but will not appear on the final layout. Next, a spacer material is deposited around the boundary of the mandrels as shown in the slashed area in (c). The mandrels will then be removed as shown in (d). After that, the second mask, trim mask shown in the green area, will be applied to block the undesired layout region. A metal filling process will then fill the white area so that the final layout in (e) is obtained. We call pattern A a *mandrel pattern* since it is defined by the mandrel mask, and pattern B a *non-mandrel pattern*.

To achieve a valid layout decomposition, all patterns on the mandrel mask and the trim mask must satisfy the minimum spacing S_{dp}. One issue with SADP process is that the trim mask may not perfectly aligned to the mandrel mask. Consequently, the overlay error occurs and it may cause line-end or CD variation.

B. Merge-and-cut Technique

The layout decomposition problem is usually formulated as a two-coloring problem, where conflicting patterns must be assigned different colors. One color will be defined by mandrel pattens, as A in Fig. 1, while the other will be defined by non-mandrel patterns, as B. A two-coloring result of Fig. 2(a) is shown in 2(b).

The challenge of SADP layout decomposition is that two-coloring method may not necessarily avoid all conflicts. To further eliminate conflicts, *merge-and-cut* technique is utilized
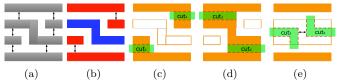
[11, 14] to merge two conflicting patterns and then trim out the unwanted part by the trim mask. Fig. 2(b) shows two conflicts remaining after two-coloring, and thus we cannot generate those patterns by the mandrel and trim mask directly. Fig. 2(c)~(e) show possible merge-and-cut solutions by merging two conflicting patterns and then cutting the unwanted area by the *cutting patterns* $cut_1 \sim cut_6$ defined by the trim mask. The pattern boundaries that directly touch the trim patterns would have potential overlay error; meanwhile, cutting patterns cannot violate the minimum spacing S_{dp} such as (e). Therefore, merge-and-cut solutions should be selected appropriately such that the cutting boundaries/overlay are as small as possible.

III. GRAPH FORMULATION WITH EMBEDDED SADP CONSTRAINTS AND CONFLICT SOLVING SOLUTIONS

Our graph formulation, *face graph*, is constructed by the flow shown in Fig. 3, and is explained in the following.
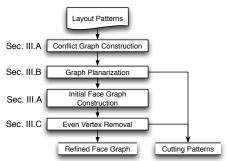

Fig. 3. Face graph construction flow.

A. Conflict Graph and Face Graph Construction

Given a 2D layout, we construct a conflict graph $G_c = (V_c, E_c)$ to express the relationship among layout patterns. Each vertex $v_c \in V_c$ represents a pattern, and each edge $e_c \in E_c$ is constructed when the distance between two patterns is less than S_{dp}. Fig. 5(a) show the conflict graph of the layout in Fig. 4(a). It has been shown that in two-coloring problem, a conflict occurs only when there is an odd cycle in the conflict graph [15]. To achieve a valid SADP layout decomposition, we have to eliminate all odd cycles in G_c.

Based on G_c, we define its dual graph, face graph $G_f = (V_f, E_f)$ where $V_f = V_{face} \cup V_{dummy}$. A vertex $v_{face} \in V_{face}$ corresponds to a face in G_c except the exterior face, and a *dummy vertex* v_{dummy} is created for each merge-and-cut candidate of a conflict. An edge $e_f \in E_f$ connects v_{face} and v_{dummy} if v_{dummy} is the solution candidate to solve the corresponding conflict of v_{face}. We call v_{face} as an *even (odd) vertex* if it corresponds to an even (odd) face in G_c. The initial face graph of the layout in Fig. 4(a) is shown in red in Fig. 5(b).
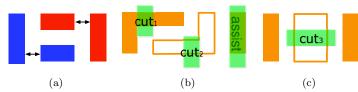
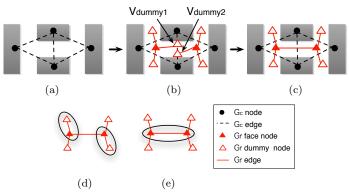Fig. 4. (a) Target layout. (b)(c) Merge-and-cut solutions.

Fig. 5. (a)(b)(c) Conflict graph (black) and face graph (red) example. (d)(e) Matching results.

For adjacent odd vertices v_{f1} and v_{f2}, they may share the same merge-and-cut candidates. For example, v_{dummy1} and v_{dummy2} in Fig. 5(b) both refer to cut_3 in Fig. 4(c). In the case where $e_{f1} = (v_{f1}, v_{dummy1})$ and $e_{f2} = (v_{f2}, v_{dummy2})$ refer to the same merge-and-cut candidate, we combine e_{f1} and e_{f2}, and remove v_{dummy1} and v_{dummy2}, as shown in Fig. 5(c).

B. Conflict Graph Planarization

It has been shown in [16,17] that the planarity of the conflict graph is based on the setting of S_{dp}. The conflict graph is planar only if Eq. (1) is satisfied:

$$\left\{ \begin{array}{ll} S_{dp} < 2 \times S_{min} & \text{in the Manhattan distance} \\ S_{dp} < \sqrt{2} \times S_{min} & \text{in the Euclidean distance} \end{array} \right. \quad (1)$$

, where S_{min} is the minimum spacing between patterns on the layout. In the case that Eq. (1) is violated, we need to planarize G_c since G_f cannot be constructed based on a non-planar graph.

If a conflict graph G_c is highly non-planar, it implies that several patterns conflict with multiple patterns. Therefore it is less possible to find a valid DPL decomposition. Be assuming that the non-planar cases in the give layout is limited, we apply the following heuristic to solve the non-planar subgraph. In a non-planar graph G_c, assume $e_1 \in E_c$ and $e_2 \in E_c$ cross each other, we eliminate one of the two edges by merging their connected vertices. Conceptually, this planarization means we force two patterns to be merged to prevent a non-planar case. In order to minimize the overlay error and EBL cost of merging two patterns, the edge with smaller merging cost (defined in Sec. IV-B) will be eliminated.

C. Even Vertex Removal for Face Graph

Given an edge $e_f = (v_{face}, v_{dummy}) \in G_f$, it implies that we can use the merge-and-cut candidate e_f to reduce the degree of the corresponding face of v_{face} by one. Since V_f contains vertices from all faces, our merge-and-cut candidate may either make an odd face become an even face (meaning the conflict

---

Algorithm 1 RemoveEvenVertex

1: $F_{odd} \leftarrow$ all odd faces in G_c
2: **for all** $f \in F_{odd}$ **do**
3: $\quad \Gamma(f)_{even} \leftarrow$ even adjacent faces of f
4: $\quad \Gamma(f)_{odd} \leftarrow$ odd adjacent faces of f
5: $\quad$ **if** $\Gamma(f)_{even} \neq \phi$ **then**
6: $\quad\quad$ Remove $v_i \in G_f, \forall f_i \in \Gamma(f)_{even}$
$\quad\quad\quad$ where v_i is the corresponding face vertex of f_i
7: $\quad$ **else**
8: $\quad\quad$ Solve f by the min-cost merge-and-cut candidate
$\quad\quad\quad$ as Sec. IV-B
9: $\quad$ **end if**
10: **end for**

---

is solved), or make an even face become an odd face (meaning a new conflict is introduced). Because applying merge-and-cut increases the risk of overlay error on the cutting boundaries, we would like to minimize new conflicts introduced by merge-and-cut. With this motivation, we apply a vertex removal heuristic in Algorithm 1 to greedily remove even vertices in G_f.

IV. LAYOUT DECOMPOSITION WITH SADP AND COMPLEMENTARY EBL

A. SADP with Complementary E-beam Lithography

A limitation of applying the merge-and-cut technique with the trim mask is that the distance between cutting patterns may violate the minimum DPL spacing S_{dp}. For example, the solution in Fig. 2(e) requires two cutting patterns cut_5 and cut_6, which actually conflict with each other.

EBL enables smaller feature width and spacing, and thus it achieve better pattern quality and design flexibility than SADP. However, EBL throughput is its biggest bottleneck as the write time is determined by the number of e-beam shots. Therefore, extensive use of e-beam cutting is not practical for manufacturing.

We adopt the conventional e-beam system where e-beam shots are variable-shaped (rectangular) beams (VSB). Cutting patterns formed by VSB require layout fracturing, meaning the patterns are decomposed into non-overlapping e-beam shots/rectangulars.

B. Min-Cost Matching based Conflict Elimination

In SADP layout decomposition problem, our objective is to eliminate conflicts with minimal overlay error and EBL utilization. We first explain our min-cost matching based conflict elimination algorithm on the *face graph*. Then we discuss how to utilize this algorithm for the overlay and EBL co-optimization, including a post processing based approach (Sec. IV-B.1) and a simultaneous optimization (Sec. IV-B.2).

The face graph defined in Sec. III has the following property:

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

According to Property A, we can solve a conflict corresponding to an odd vertex by selecting one of its connecting edges $e_f \in E_f$. It is obvious that selecting more than one e_f for an odd vertex is unnecessary because a new conflict would be introduced. Therefore, we seek to find one merge-and-cut

---

Algorithm 2 Hybrid-Post

Input: $G_f = (V_f, E_f)$ // Sec. III
Output: $P_{cut} = P_{optical} \cup P_{ebl}$, with the objective of
    minimizing $\sum_{e_i \in P_{optical}} cost_{e_i}$ and $\sum_{e_j \in P_{ebl}} N_{shot}(e_j)$
  1: AssignCost_SADP(E_f)
  2: $P_{allcuts} \leftarrow$ RunMatching(G_f)
  3: $G_{mc} \leftarrow$ ConstructConflictGraph($P_{allcuts}$)
  4: $P_{optical} \leftarrow MIS(G_{mc})$
  5: $P_{ebl} \leftarrow P_{allcuts} - P_{optical}$

Algorithm 3 Hybrid-Sim

Input: $G_f = (V_f, E_f)$ // Sec. III
Output: $P_{cut} = P_{optical} \cup P_{ebl}$, with the objective of
    minimizing $\sum_{e_i \in P_{optical}} cost_{e_i}$ and $\sum_{e_j \in P_{ebl}} N_{shot}(e_j)$
  1: AssignCost_SADP(E_f)
  2: $P_{conf} = \Phi$
  3: **repeat**
  4:    $P_{optical}, P_{ebl} \leftarrow$ RunMatching(G_f)
  5:    $P_{conf} \leftarrow$ ValidateCut($P_{optical}$)
  6:    SubstituteEBL(P_{conf})
  7: **until** $P_{conf} == \Phi$

---

solution for each conflict, and we formulate the conflict elimination problem as the matching problem, where each match corresponds to solving a conflict. For example, Fig. 5(d) and (e) show two different matching results which corresponds to the final masks shown in Fig. 4(b) and (c), respectively.

B.1   Post Processing Based Conflict Elimination

Since the trim mask itself can be used for the merge-and-cut technique to solve conflicts, we can view EBL cutting as a back-up solution during conflict elimination. We propose a two-stage approach for overlay error and e-beam optimization. First, we solve all conflicts by applying the min-cost matching algorithm on G_f, then we assign the obtained cutting patterns to the trim mask and e-beam shots according to SADP constraint S_{dp}. The approach flow is shown in Algorithm 2, where $N_{shot}(e)$ represent the required number of e-beam shots for the merge-and-cut candidate corresponds to e.

In the beginning of Algorithm 2, G_f is constructed based on Sec. III. Since EBL is not considered in the first stage, we model the edge cost simply by the SADP overlay error according to Eq. (2) in Line 1.

$$cost_e = L_{boundary}(e) \quad \forall e \in E_f \tag{2}$$

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

We then solve all conflicts by merge-and cut technique (Line 2). The cutting patterns $P_{allcuts}$ with the minimum overlay can be obtained by performing the min-cost matching algorithm in Line 2. However, there may exist conflicts among the cutting patterns because of the S_{dp} constraint. With e-beam available, we can carefully select a subset of cutting patterns $P_{optical}$ that do not violate S_{dp}, and let the rest of the cutting patterns P_{ebl} formed by EBL. We first construct a conflict graph G_{mc} for $P_{allcuts}$ in Line 3 to check if there is any conflict among $P_{allcuts}$. In order to minimize e-beam shot utilization, we apply maximal independent set (MIS) algorithm on G_{mc} to obtain the maximal number of valid patterns for $P_{optical}$ (Line 4), and assign the rest of the cutting patterns to be done by EBL (Line 5). Based on the property of MIS, the S_{dp} constraint is guaranteed to be satisfied for $P_{optical}$.

B.2   Conflict Elimination with Simultaneous Overlay and EBL Throughput Optimization

Although the approach in Sec. IV-B.1 can successfully solve conflicts with hybrid SADP and EBL, it only considers EBL in the last stage and does not include e-beam optimization when finding the min-cost matching. To further improve the decomposition result, we propose a simultaneous overlay error and EBL throughput optimization as shown in Algorithm 3. The main idea of the algorithm is to start from a restricted solution

space and gradually increase the solution space with more EBL merge-and-cut candidates until we find a valid solution. Based on the algorithm, only necessary e-beam candidates are considered, and the matching algorithm can simultaneously optimize SADP overlay error and e-beam utilization.

In the beginning of Algorithm 3, G_f is constructed based on Sec. III. All edges are initialized as optical cuts with the cost defined by Eq. (2). We then iteratively perform min-cost matching algorithm in Line 4 to find the cutting patterns. Since we may obtain conflicts among some optical cutting patterns in Line 5, we substitute those conflicting optical cuts by EBL cuts in Line 6 with the cost function in Eq. (3).

$$cost_e = C_{ebl} \times N_{shot}(e) \quad \forall e \in P_{conf} \tag{3}$$

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

In our implementation, we set C_{ebl} sufficiently large than the cost of any optical cut, such that optical cuts are always preferred than EBL cuts. By including both overlay and e-beam cost with Eq. (2) and Eq. (3), our min-cost matching solution can minimize the overlay error and e-beam utilization simultaneously. Note that we assume rectangular beam shape is applied. For example, an L-shaped pattern requires at least two beam shots. In addition, there are minimum size and maximum size limitation for beam shape, which would also affect the number of shots N_{shot} of each merge-and-cut candidate. Because a merge-and-cut candidate is formed between patterns with half-pitch width, generally the minimum shape constraint would not be violated; and the maximum shape constraint would apply for long cutting patterns.

V. Overlay-aware Conflict Minimization for pure SADP

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

A. Adapted Face Graph for Conflicting Cuts

In the face graph defined in Sec. III, the merge-and-cut candidates may conflict each other. Therefore, when applying the min-cost matching algorithm to solve conflicts as explained in Sec. IV-B, we may obtain conflicting cutting patterns. These
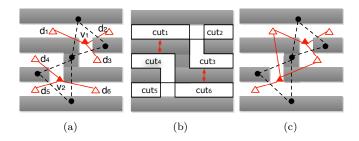
Fig. 6. Face graph adapted for conflicting cuts. (a) Conflict graph (black) and Face graph (red). (b) Cutting patterns that cannot co-exist are indicated by arrows. (c) Adapted face graph without conflicting cuts.

conflicts must be prevented in pure SADP layout decomposition.

Fig. 6(a) shows the layout patterns and its corresponding conflict and face graph. Two conflicts v_1 and v_2 are discovered because they forms odd cycles in the conflict graph. The merge-and-cut candidates of v_1 and v_2 are shown by $cut_1 \sim cut_3$ and $cut_4 \sim cut_6$ in Fig. 6(b), respectively. It can be seen that if we select cut_1 and cut_4 to solve the two conflicts, the solution would not be valid because the two cuts are too close to be fabricated on the same mask. Similarly, cut_3 and cut_6 cannot co-exist.

To add the conflicting cuts information into our face graph, we check all merge-and-cut candidates by traversing all edges, and make conflicting edges connect to the same dummy vertex. Finally, we can obtain an adapted face graph G'_f with the conflicting cuts information embedded. As shown in Fig. 6(c), after this graph adaption, d_1 and d_4 is merged, and so is d_3 and d_6.

B. Matching based Conflict Minimization

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

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

Property B guarantees that the matching algorithm would only select one edge that covers the same dummy vertex. This ensures that S_{dp} rule is satisfied for the cutting patterns. Besides, Property A in Sec. IV- B still holds for the adapted face graph. Consequently, by modeling the edge cost of G'_f with the desired SADP cost, we can minimize conflicts with the min-cost matching algorithm presented in Sec. IV- B. Here our objective is to minimize overlay error introduced by cutting patterns, therefore, Eq. (2) is adopted in the matching problem. This approach allows simultaneous optimization for overlay minimization and conflict minimization for pure SADP layout decomposition.

VI. EXPERIMENTAL RESULTS

The proposed algorithms are implemented in C++ and tested on Intel platform with 2.66 GHz CPU and 4G memory. We synthesize OpenSPARC T1 designs with Nangate 45nm standard cell library [18], and perform placement and routing with Cadence SOC Encounter [19] to generate the layouts. These layouts are then scaled down for 22nm technology node. For simplicity, we assume the sizes of the minimum pattern width, spacing, and spacer width are 50nm, and make the

corresponding adjustment for the benchmark. The double patterning spacing S_{dp} is set large enough to introduce conflicts to evaluate the performance of our algorithms. Table I shows the statistics of five designs with the number of 2D patterns (#Polygon) and the initial coloring conflicts (#Conf) before applying our approaches, where #Conf is obtained based on the two-coloring result.

A. Overlay-aware Layout Decomposition for SADP

We first apply the proposed approach in Sec. V to solve conflicts with the trim mask for conflict and overlay error minimization. Because the existing approaches for SADP layout decompositions [10–13] are performed for two-colorable cases, no solution would be generated for comparison on designs with conflicts. Although layout perturbation [20] can be applied to solve native conflicts, we do not allow layout change in our problem. An alternative to solve this problem is to minimize the total number of cutting patterns, by which we expect less overlay error and unsolved conflicts because less cutting patterns compete for the mask resource. As a baseline, we implement this alternative (SADP-Cut) by replacing Eq. (2) with Eq. (4) and compare it with the proposed overlay-aware layout decomposition (SADP-OV).

$$cost_e = 1 \quad \forall e \in E_f \qquad (4)$$

Table I shows the results after layout decomposition in terms of the remaining conflicts ($\#Conf_{rem}$), the total length of the overlay-risky boundaries touched by the trim mask ($Bndy_{ov}$), and the CPU time. It can be seen that the merge-and-cut technique is not sufficient to solve all conflicts because of the resolution limit of the trim mask. Compared with the baseline, SADP-OV successfully reduces the overlay error, whose effect can be represented by the length of the cutting boundaries. On average, the overlay error can be reduced by 28.36% with SADP-OV. Although there are still outstanding conflicts that cannot be resolved, our approach successfully resolve more than 95% of the initial conflicts, showing that merge-and-cut is promising for SADP layout decomposition. Note that our benchmarks are not targeting at any specific lithography process and thus are not SADP-friendly designs. By properly designing the layout for SADP or specify lithography-aware rules in early design stages, it would be easier to solve conflicts by our approach.

B. Overlay and EBL Throughput Co-optimization for SADP with Complementary EBL

By adopting complementary EBL, the conflicts that cannot be handled in Sec. VI- A can be solved. We compare the layout decomposition results when applying the two conflict elimination approaches, Hybrid-Post and Hybrid-Sim. Because Hybrid-Post is a two-stage approach, we would like to study how the result in the first stage affects the final result. Therefore, two versions of Hybrid-Post are implemented, one perform the min-cost matching algorithm based on Eq. (2) (Hybrid-Post-OV), while the other perform the min-cost matching algorithm based on Eq. (4) (Hybrid-Post-Cut).

The results are shown in Table II, where $\#VSB$ refers to the total number of variable shaped beams. Note that the cutting patterns are 2-dimensional and thus a conflict may require more than one VSB to solve. With complementary EBL, all conflicts in our benchmark are solved. It can be seen that Hybrid-Post-Cut requires a large number of VSB because it

TABLE I
Overlay-aware layout decomposition for SADP.

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

TABLE II
Layout decomposition for Overlay and EBL throughput co-optimization

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

leaves more unsolved conflicts before applying e-beams. The simultaneous optimization Hybrid-Sim outperforms the two post-processing based approaches, which reduces VSB utilization by 69% while achieving comparable overlay error minimization with Hybrid-Post-OV.

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

VII. Conclusion

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

Acknowledgment

References

[1] David Z. Pan, Bei Yu, and Jhih-Rong Gao. Design for manufacturing with emerging nanolithography. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2013.

[2] Yukiyasu Arisawa, Hajime Aoyama, Taiga Uno, and Toshihiko Tanaka. EUV flare correction for the half-pitch 22nm node. In *Proc. of SPIE*, 2010.

[3] Aki Fujimur. Beyond light: The growing importance of E-Beam. In *Proc. Int. Conf. on Computer Aided Design*, 2009.

[4] Chris Bencher, Jeffrey Smith, Liyan Miao, Cathy Cai, Yongmei Chen, Joy Y Cheng, Daniel P Sanders, Melia Tjio, Hoa D Truong, Steven Holmes, and William D Hinsberg. Self-assembly patterning for sub-15nm half-pitch: a transition from lab to fab. In *Proc. of SPIE*, March 2011.

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

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

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

[8] Kenichi Oyama, Eiichi Nishimura, Masato Kushibiki, Kazuhide Hasebe, Shigeru Nakajima, and et al. The important challenge to extend spacer DP process towards 22nm and beyond. In *Proc. of SPIE*.

[9] Yuelin Du, Hongbo Zhang, M.D.F. Wong, and Kai-Yuan Chao. Hybrid lithography optimization with E-Beam and immersion processes for 16nm 1D gridded design. In *Proc. Asia and South Pacific Design Automation Conf.*, 2012.

[10] Minoo Mirsaeedi, J. Andres Torres, and Mohab Anis. Self-aligned double patterning (SADP) layout decomposition. In *Proc. Int. Symp. on Quality Electronic Design*, 2011.

[11] Yongchan Ban, K. Lucas, and D. Z. Pan. Flexible 2D layout decomposition framework for spacer-type double pattering lithography. In *Proc. Design Automation Conf.*, June 2011.

[12] Hongbo Zhang, Yuelin Du, Martin D. F. Wong, and Rasit Topaloglu. Self-aligned double patterning decomposition for overlay minimization and hot spot detection. In *Proc. Design Automation Conf.*, 2011.

[13] Zigang Xiao, Hongbo Zhang, Yuelin Du, and Martin D. F. Wong. A polynomial time exact algorithm for overlay-resist self-aligned double patterning (SADP) layout decomposition. In *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2013.

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

[15] Andrew B. Kahng, Chul-Hong Park, Xu Xu, and Hailong Yao. Layout decomposition for double patterning lithography. In *Proc. Int. Conf. on Computer Aided Design*, November 2008.

[16] Yue Xu and Chris Chu. A matching based decomposer for double patterning lithography. In *Proc. Int. Symp. on Physical Design*, 2010.

[17] Shao-Yun Fang, Szu-Yu Chen, and Yao-Wen Chang. Native-conflict and stitch-aware wire perturbation for double patterning technology. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2012.

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

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

[20] R.S. Ghaida, K.B. Agarwal, S.R. Nassif, Xin Yuan, L.W. Liebmann, and P. Gupta. Layout decomposition and legalization for double-patterning technology. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2013.