

Cut Mask Optimization for Multi-Patterning Directed Self-Assembly Lithography

Wachirawit Ponghiran, Seongbo Shim, and Youngsoo Shin
School of Electrical Engineering, KAIST
Daejeon 34141, Korea

Abstract—Line-end cut process has been used to create very fine metal wires in sub-14nm technology. Cut patterns split regular line patterns into a number of wire segments with some segments being used as actual routing wires. In sub-7nm technology, cuts are smaller than optical resolution limit, and a directed self-assembly lithography with multiple patterning (MP-DSAL) is considered as a patterning solution. We address cut mask optimization problem for MP-DSAL, in which cut locations are determined in such a way that cuts are grouped into manufacturable clusters and assigned to one of masks without MP coloring conflicts; minimizing wire extensions is also pursued in the process. Only a restricted version of this problem has been addressed before while we do not assume any such restrictions. The problem is formulated as ILP first, and a fast heuristic algorithm is also proposed for application to larger circuits. Experimental results indicate that the ILP can remove all coloring conflicts, and reduce total wire extensions by 93% on average compared to those obtained by the restricted approach. Heuristic achieves a similar result with less than 1% of coloring conflicts and 91% reduction in total wire extensions.

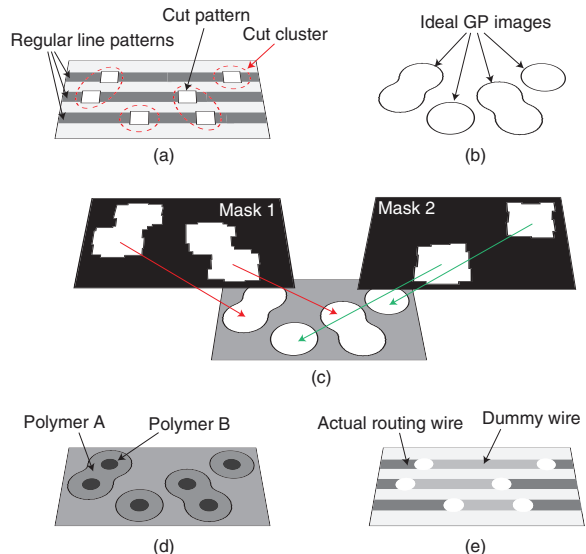


Fig. 1. Line-end cut process with MP-DSAL: (a) regular line patterns with cut clusters, (b) ideal GP images of cut clusters, (c) multi-patterning to create GPs on a wafer, (d) GPs filled with BCP, and (e) routing- and dummy-wires created after polymer B is etched away.

I. INTRODUCTION

As technology node scales down to sub-7nm, printing small features becomes challenging task for traditional optical lithography. MP-DSAL has recently received much attention as an alternative patterning solution due to its low manufacturing cost and high printing resolution. MP-DSAL is considered as a viable technique for printing contact and via layers [1]–[3], but its application for printing line-end cuts (or cuts for short) has also been studied [4].

In line-end cut process, regular line patterns are first created on a wafer through self-aligned double patterning (SADP) [5]; they are then split into multiple wire segments by cuts. Some segments are used as actual routing wires while the remainders become dummies and serve no function. Process to create cut patterns with MP-DSAL is as follows: cuts that are physically close are clustered (Fig. 1(a)) and contour that surrounds each cluster, called a guide pattern (GP) image, is synthesized (Fig. 1(b)). If a distance between two GP images is smaller than optical resolution limit, they are patterned on different masks (Fig. 1(c)) and actual GPs are created on a wafer by processing each mask one by one. GPs are filled with block copolymers (BCPs), which align by themselves due to forces between BCPs and GP walls during heating and annealing process (Fig. 1(d)). After one type of polymer (polymer B) is etched down to the underlying metal layer, regular line patterns are split into actual routing and dummy wires (Fig. 1(e)).

Clustering nearby cuts is not arbitrary. A large and complex GP image is hard to manufacture during DSA process [6], so GP can contain only a limited number of cuts and its shape should be simple. In addition, GPs have to be assigned to one of masks without coloring conflicts. This makes cut clustering problem with mask assignment very difficult. In fact, a similar version of problem in contact and via layers has been shown to be NP-complete [1], [7].

A. Motivation and Contributions

Unlike contacts and vias whose locations are fixed after layout design, cuts can be relocated to avoid coloring conflicts and forming unmanufacturable clusters. However, relocating cuts results in wire extensions as shown in Fig. 2(a). This may negatively affect circuit timing due to increased capacitance. The goal of cut mask optimization for MP-DSAL, which we address in this paper, is to determine cut locations together with cut clustering and mask assignment, in such a way that all clusters are manufacturable with smallest coloring conflicts and wire extensions are minimized. We aim to minimize coloring conflicts instead of searching for conflict-free solution

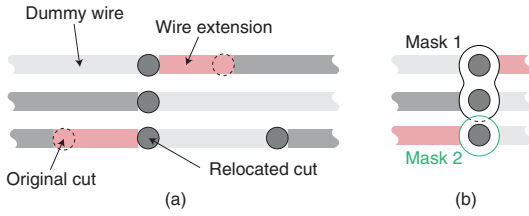


Fig. 2. (a) Wire extensions after some cuts are relocated, and (b) overlap between clusters that are assigned to different masks.

because small number of conflicts can be manually fixed by designers.

A restricted and simplified version of this problem has been previously addressed [4], but its solution is impractical. This is because the approach performs cut mask optimization problem for basic DSAL (without MP) independently for cuts on every three consecutive metal tracks, and implicitly assigns cut clusters on these tracks to one mask. By selecting different masks for adjacent metal track groups, coloring conflict across each group is guaranteed to be free. However, overlap may occur between clusters assigned to different masks is as shown in Fig. 2(b). In addition, assigning only one mask to clusters on each track group would limit solution space and produce result that is far from optimal (as demonstrated in Section IV through experiments).

In this paper, we address the general version of cut mask optimization problem for MP-DSAL; key contributions are summarized as follows:

- Cut mask optimization for MP-DSAL is formulated as ILP, which serves as a foundation for developing heuristic algorithm and is also used as a reference for comparison.
- A fast heuristic algorithm is developed for application to large practical circuits.

The remainder of this paper is organized as follows. Section II briefly discusses critical cut distances in MP-DSAL, and the impact of wire extensions on circuit timing. Section III presents the ILP formulation along with the heuristic algorithm. Our proposed methods are demonstrated and compared with the existing approach in Section IV. The paper is summarized in Section V.

II. PRELIMINARIES

A. Critical Cut Distances in MP-DSAL

Suppose p denotes a center-to-center distance or pitch between two cuts; minimum pitch supported by DSAL is denoted by p_{dsa}^- ; maximum length that BCP can stretch is denoted by p_{dsa}^+ . For any two cuts to be clustered together, their pitch must be at least p_{dsa}^- but no more than p_{dsa}^+ ; this range of p is marked as DSA in Fig. 3. If their pitch is larger than p_{mp} , which is a minimum pitch supported by multi-patterning, cuts can be assigned to different clusters whose GPs are patterned on different masks. This range of p is denoted as MP in Fig. 3. Hence, if a pitch between two cuts

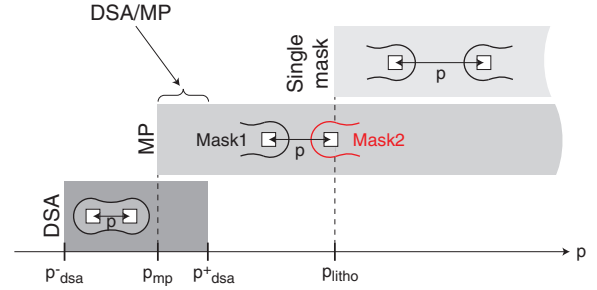


Fig. 3. Cut clustering and mask assignment for various cut pitches.

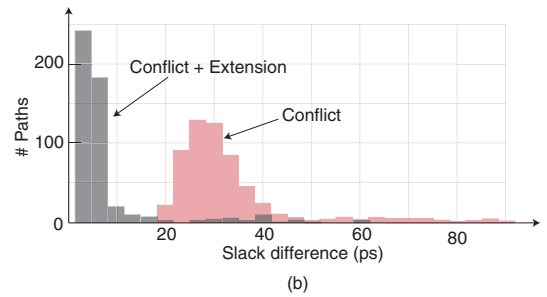
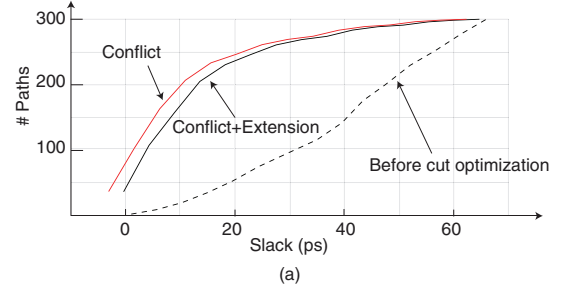


Fig. 4. (a) Slack histograms of test circuit before and after cut mask optimizations, and (b) slack changes of critical paths after cut mask optimizations.

lies in $p_{mp} \leq p \leq p_{dsa}^+$ (marked as DSA/MP), they can be either assigned to the same cluster or assigned to different clusters whose GPs are patterned on different masks. Lastly, if a pitch between two cuts is larger than p_{litho} , which is a minimum pitch supported by traditional optical lithography, cuts can be freely assigned to any different clusters. This range of p is marked as Single mask in Fig. 3.

The value of p_{dsa}^- and p_{dsa}^+ are determined by the length of BCP [8] that is employed. The value of p_{mp} and p_{litho} are determined from the BCP length as well as optical resolution limit, which in turn is affected by lithography settings such as wavelength of light source, NA, and illumination shape.

B. Wire Extension: Impact on Timing

Wire extension has been considered as a popular cost function in many existing studies of cut mask optimization [4], [9]–[11]. Nevertheless, its impact on circuit timing has never been discussed in quantitative fashion, which we try in this section using 28nm commercial library.

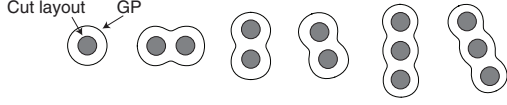


Fig. 5. GPs that are assumed to be manufacturable in the paper.

We obtain two optimal solutions of cut mask optimization problem using ILP (explained in Section III). The first solution (denoted by Conflict+Extension in Fig. 4) has least number of coloring conflicts and total wire extensions. The second solution (denoted by Conflict) has the minimum number of color conflicts but wire extensions are just limited below a certain value.

Slack histograms of test circuit (spi) before and after cut mask optimization are shown in Fig. 4(a). There are many paths with slack reductions after the optimizations. To justify the impact of wire extensions, we measure slack change of critical paths that originally have slack less than 50ps. As shown in Fig. 4(b), slack of critical paths in Conflict+Extension reduces by 15ps on average (and up to 72ps). Slack of critical paths in Conflict, on the other hand, reduces by 42ps on average (and up to 97ps). This large difference in slack between two optimal solutions may lead to significant timing degradation, and is also observed in other test circuits. Thus, it is reasonable to consider the total wire extensions as a cost function in cut mask optimization problem.

III. MP-DSAL CUT MASK OPTIMIZATION

Cut mask optimization for MP-DSAL determines the location of all cuts while each cut is assigned to a cluster (for DSAL) which is then assigned to a mask (for MP). It guarantees that (1) circuit connectivity remains the same, (2) total wire extensions are minimized, and (3) MP coloring conflicts are kept at a minimum. In this section, we first present ILP formulation for this problem, and move on to propose a heuristic algorithm. Note that only clusters corresponding to GPs with zero defect probability (i.e. GPs with linearly aligned cuts as illustrated in Fig. 5) are assumed.

A. ILP Formulation

We define a gap (g_i) as a space between adjacent actual routing wires on the same metal track as shown in Fig. 6(a). g_i is then associated with two cuts, namely left cut (denoted c_{2i}) at position (x_{2i}, y_{2i}) and right cut (denoted c_{2i+1}) at position (x_{2i+1}, y_{2i+1}) . Without loss of generality, we assume that wires are routed on a horizontal direction. After a cut mask optimization, horizontal location of left and right cut are changed to \hat{x}_{2i} and \hat{x}_{2i+1} respectively while vertical location of cuts remains the same. This leads to amount of wire extension $\hat{x}_{2i} - x_{2i}$ on the left and $x_{2i+1} - \hat{x}_{2i+1}$ on the right as illustrated in Fig. 6(b). The rest of notations used in ILP are summarized in Table I.

TABLE I
NOTATIONS OF THE ILP FORMULATION

g_i	i -th gap
c_i	i -th cut
(x_i, y_i)	Original location of cut
\hat{x}_i	Horizontal location of cut after optimization
G_{ik}	1 if c_i is assigned to k -th manufacturable cluster
M_{ik}	1 if the cluster of c_i is assigned to k -th mask
C_{ij}	1 if the cluster of c_i and c_j have a coloring conflict; 0 otherwise
O_{ij}	1 if (\hat{x}_i, y_i) equals to (\hat{x}_j, y_j) ; 0 otherwise
S_{ij}	1 if c_i and c_j are on the same cluster; 0 otherwise
M_{ij}	1 if the cluster of c_i and c_j are on the same mask; 0 otherwise
$p_{min}(i, j)$	Minimum pitch between c_i and c_j
$h_{dsa}^-(i, j)$	Minimum horizontal pitch between c_i and c_j to be on the same cluster
$h_{dsa}^+(i, j)$	Maximum horizontal pitch between c_i and c_j to be on the same cluster
$h_{mp}(i, j)$	Minimum horizontal pitch between c_i and c_j to be on different clusters patterned on different masks
$h_{litho}(i, j)$	Minimum horizontal pitch between c_i and c_j to be on different clusters patterned on the same mask
m_{ik}	k -th bit of mask index of cluster that c_i belongs
M_{ijk}	1 if k -th bit of mask index of cluster that c_i and c_j belong are identical; 0 otherwise
W	Large constant

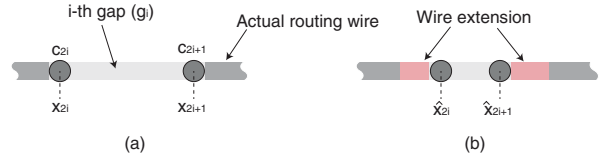


Fig. 6. A gap and its corresponding cuts on the left and right (a) before and (b) after the cuts are relocated.

The cut optimization problem for MP-DSAL is to determine the value of \hat{x}_i , G_{ik} , and M_{ik} with objective to minimize total wire extensions and coloring conflicts.

Minimize

$$\sum_i [\hat{x}_{2i} - x_{2i} + x_{2i+1} - \hat{x}_{2i+1}] + W \left[\sum_{i,j} C_{ij} \right] \quad (1)$$

subject to:

$$\sum_k G_{ik} = 1, \quad \forall c_i \quad (2)$$

$$x_{2i} \leq \hat{x}_{2i} \leq \hat{x}_{2i+1} \leq x_{2i+1}, \quad \forall g_i \quad (3)$$

$$O_{ij} \leq S_{ij} \leq M_{ij}, \quad \forall c_i \text{ and } c_j (i \neq j) \quad (4)$$

for all c_i and c_j such that $\hat{x}_i \leq \hat{x}_j$ and $p_{min}(i, j) \leq p_{dsa}^+$,

$$\hat{x}_j - \hat{x}_i \geq h_{dsa}^-(i, j) - W[(1 - S_{ij}) + O_{ij}], \quad (5)$$

$$\hat{x}_j - \hat{x}_i \leq h_{dsa}^+(i, j) + W[(1 - S_{ij}) + O_{ij}], \quad (6)$$

for all c_i and c_j such that $\hat{x}_i \leq \hat{x}_j$ and $p_{min}(i, j) \leq p_{mp}$,

$$\hat{x}_j - \hat{x}_i \geq h_{mp}(i, j) - W[S_{ij} + M_{ij}], \quad (7)$$

for all c_i and c_j such that $\hat{x}_i \leq \hat{x}_j$ and $p_{min}(i, j) \leq p_{litho}$,

$$\hat{x}_j - \hat{x}_i \geq h_{litho}(i, j) - W[S_{ij} + (1 - M_{ij}) + C_{ij}], \quad (8)$$

for all c_i and c_j such that $i \neq j$,

$$M_{ijk} \geq 1 - m_{ik} - m_{jk}, \quad \forall k \in \{1, 2\} \quad (9a)$$

$$M_{ijk} \leq 1 - m_{ik} + m_{jk}, \quad \forall k \in \{1, 2\} \quad (9b)$$

$$M_{ijk} \leq 1 + m_{ik} - m_{jk}, \quad \forall k \in \{1, 2\} \quad (9c)$$

$$M_{ijk} \geq m_{ik} + m_{jk} - 1, \quad \forall k \in \{1, 2\} \quad (9d)$$

$$2M_{ij} \leq M_{ij1} + M_{ij2}, \quad (10a)$$

$$M_{ij} \geq M_{ij1} + M_{ij2} - 1. \quad (10b)$$

Inequality (2) restricts that each cut is assigned to only one cluster. Inequality (3) specifies feasible location of cuts so that circuit connectivity remains the same. Inequality (4) guarantees that two cuts at the same location must be on the same cluster. Cuts on the same cluster must also be on the same mask. Inequalities (5)-(6) restricts that two cuts can be clustered together if their pitch is within the range of DSA (see Fig. 3). $h_{dsa}^-(i, j)$ is obtained by $\sqrt{(p_{dsa}^-)^2 - (y_i - y_j)^2}$, and $h_{dsa}^+(i, j)$ is obtained similarly. For any two cuts on different clusters patterned on different masks, inequality (7) restricts that their pitch is in the range of MP (see Fig. 3). If two cuts are on different clusters patterned on the same mask, inequality (8) ensures that their pitch is in the range of Single Mask (see Fig. 3). Otherwise, their clusters has a coloring conflict. Note that $h_{mp}(i, j)$ and $h_{litho}(i, j)$ are obtained in similar manner to $h_{dsa}^-(i, j)$.

Assuming that upto four masks can be used, we represent the mask of each cluster with two-bit number $m_{i1}m_{i2}$ in the ILP formulation. In case of double- and triple-patterning, additional constraints $m_{i1} = 0$ and $m_{i1} + m_{i2} \leq 1$ should be added, respectively. A set of inequalities (9a)-(9d) is equivalent to XNOR operation (e.g. $M_{ij1} = m_{i1} \text{ XNOR } m_{j1}$), which sets M_{ijk} to 1 if the k -th bit of mask index of cluster that c_i and c_j belong are identical. Inequality (10a)-(10b) correspond to AND operations (i.e. $M_{ij} = M_{ij1} \text{ AND } M_{ij2}$), which determines M_{ij} .

B. Heuristic Algorithm

We propose a fast heuristic algorithm which uses divide and conquer strategy to solve the cut mask optimization for large circuits. A potential conflict graph is initially constructed to represent all possible coloring conflicts during cut mask optimization. By ignoring some edges, the potential conflict graph is partitioned into many small acyclic ones. Each acyclic sub-graph is optimized independently, and the solutions of all the acyclic sub-graphs are then combined. There may still remain some coloring conflicts due to ignored edges during a potential graph partition; these conflicts are resolved through a local adjustment.

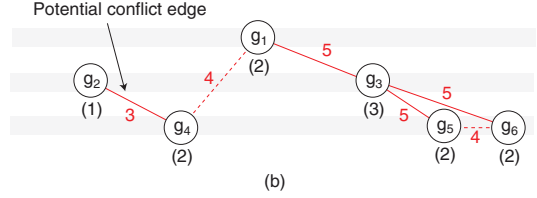
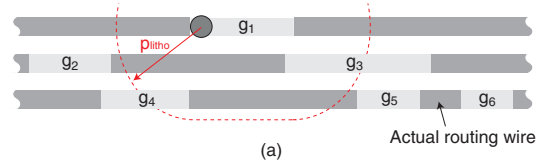


Fig. 7. (a) Example metal layout, and (b) its potential conflict graph.

Potential conflict graph construction: For a given layout, we represent each gap with a vertex as shown in Fig. 7. Potential coloring conflicts between gaps are identified based on a feasible moving region of cuts on those gaps. If gaps are closer than p_{litho} (see g_1 and g_4 in Fig. 7(a)), coloring conflict may exist between their cuts during optimization. We draw an edge to represent each potential coloring conflict as shown in Fig. 7(b).

Degree of each vertex (see the numbers in brackets) indicates how many potential coloring conflicts its corresponding gap has. To minimize potential coloring conflict of ignored edges during a graph partition, we try to group large-degree vertex and its neighbors together in the same acyclic sub-graph. To do so, we assign each edge with a weight equal to sum of the degree of the vertices incident to it (see the numbers in red). Then, partitioning is done by ignoring all edges, and consecutively including each of them in a decreasing weight order. We exclude an edge if connecting it causes a cycle, or causes a number of the vertices in a acyclic sub-graph to exceed a certain value. The size of acyclic sub-graphs are kept small in order to partition cut mask optimization of the whole layout into many small problems. When limiting a number of the vertices in acyclic sub-graphs to 4, we partition the potential conflict graph in Fig. 7(b) into 2 sub-graphs (see solid edges).

Solving acyclic sub-graph to an optimal: We convert each acyclic sub-graph into a tree by randomly picking one vertex to be a root. For instance, given the left sub-graph in Fig. 7(b), tree in Fig. 8(a) is formed by assigning g_1 to be a root. Note that each tree node represents an individual gap on a layout. We then construct another graph, called configuration graph, in which each vertex (v_i) represents a feasible configuration (location, clustering, and mask) of the cuts on each gap. Some examples of v_i corresponding to gap g_1 and g_3 are shown in Fig. 8(b). There exists an edge connecting between v_i and v_j if v_i and v_j represent configurations of the cuts on two connected gaps of a tree, and together form a valid configuration. For example, v_1 is connected to v_4 because v_1

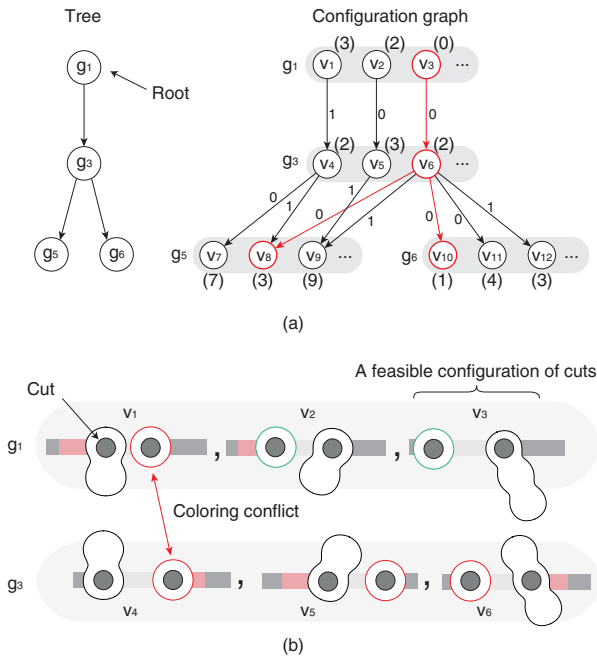


Fig. 8. (a) A tree and its configuration graph, and (b) example vertices corresponding to gap g_1 and g_3 in the tree.

represents a configuration of the cuts on gap g_1 , v_4 represents a configuration of the cuts on gap g_3 , and gap g_1 and g_3 are connected on the tree. v_1 and v_4 also have a matching cluster on the left. However, v_1 is not connected to v_5 because their left clusters are different.

Each v_i is associated with a weight equal to amount of wire extensions (see number in brackets). If any v_i and v_j are connected but have a coloring conflict, their incident edge has a weight 1 (see v_1 and v_4 in Fig. 8(b)). Otherwise, the edge has a weight 0 (see v_2 and v_5). Solution of acyclic sub-graph is obtained by finding a set of connected vertices $\{v_1, v_2, \dots, v_k\}$ where each v_k represents a configuration of the cuts on individual gaps of a tree. Example solution is colored with red in Fig. 8(a). The set of connected vertices with a minimum cost defined in (1) is thus the optimum solution of acyclic sub-graph. It can be obtained by transversing a tree in post-order ($g_5 \rightarrow g_6 \rightarrow g_3 \rightarrow g_1$) and computing minimum cost (MC) at its corresponding vertices as follows.

$$MC(v_i) = w(v_i) + \sum_{c \in C_i} \min_j [W \cdot w(e_{ij}) + MC(v_j)], \quad (11)$$

where $w(v_i)$ is the weight of vertex v_i , C_i is child node of the gap corresponding to vertex v_i , j represents all configurations of cuts on the gap c , and $w(e_{ij})$ is the weight of an edge between v_i and v_j . For illustration, the minimum cost of vertex v_6 in Fig. 8(a) is computed as $2+3+1$ or 6. 2 is a weight of vertex v_6 itself. 3 is the minimum additional cost among configurations of cuts on gap g_5 , i.e. $0+3$ (v_8) and $W+5$ (v_9). 1 is the minimum additional cost among configurations of cuts on gap g_6 , i.e. $0+1$ (v_{10}), $0+4$ (v_{11}), and $W+3$ (v_{12}).

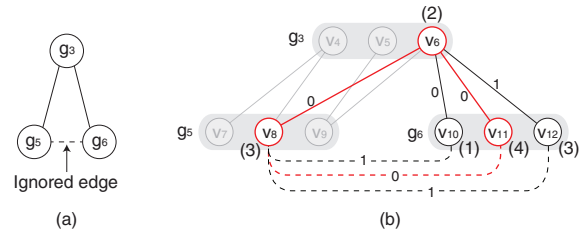


Fig. 9. (a) An ignored edge, and (b) configuration graph for a local adjustment.

Once the minimum cost of all vertices corresponding to the root gap is computed, one vertex with the smallest cost is selected. Descending vertices associated with this cost are also subsequently selected in a top-down manner, and form an optimal solution of acyclic graph.

Local adjustment: After optimizing all cuts in the previous step, we check edges that are ignored during graph partitioning for any coloring conflict. If coloring conflicts exist, we try to resolve them using a local adjustment.

Local adjustment is done by finding an alternative configuration of cuts on the conflicted gaps while assuming that configuration of other cuts are the same. Suppose that $\{v_6, v_8, v_{10}\}$ is selected as a solution in the previous step. When we consider an ignored edge $g_5 - g_6$ shown in Fig. 9(a), a coloring conflict exists. New configuration with the smallest sum of edge and vertex weights is then chosen instead. That is $\{v_6, v_8, v_{11}\}$ with total weight of 9 (see red lines in Fig. 9(b)).

IV. EXPERIMENTS

We implemented our algorithm with C++, and used GUROBI [12] as an ILP solver. Test circuits was taken from Open Cores [13] and ITC99 benchmark [14], and synthesized using 28-nm industrial library. M2 layout that we used is appropriately shrunk to follow sub-7nm design rule which we assume in this paper. Metal track pitch is 35nm, and corresponding cut size is assumed to be 18nm by 18nm. The number of cuts on test circuits ranges from 5k (b12) to 360k (des3_perf) as shown in Table II. Double patterning with ArF immersion lithography (1.35 NA) is assumed. Critical cut distances in MP-DSAL are assumed as follows: $p_{dsa}^- = 30nm$, $p_{dsa}^+ = 45nm$, $p_{mp} = 39nm$, and $p_{litho} = 87nm$.

An approach proposed in [4] is also implemented for comparison. When it is applied, many conflicts (coloring conflicts + overlap) remain because overlaps between clusters assigned to different masks are ignored. Our approaches, on the other hand, guarantee to have no overlap between clusters. ILP completed the cut mask optimization for 5 small circuits (see column 5-7). Results show that ILP can remove all coloring conflicts, and reduce total wire extensions by 93% on average compared to the approach [4] being applied. Our heuristic algorithm achieves a similar result with less than 1% coloring conflict and 91% reduction in total wire extensions (see column 8-10).

TABLE II
COMPARISON OF THREE CUT MASK OPTIMIZATION APPROACHES

Circuits	# Cuts	Approach in [4]		ILP			Heuristic		
		# Conflicts	Δ WL (mm)	# Conflicts	Δ WL (%)	Runtime (m)	# Conflicts	Δ WL (%)	Runtime (m)
b12	4,776	2,122	0.5	0	4%	14	2	11%	1
spi	10,658	4,340	1.2	0	7%	27	3	9%	1
des3_area	16,164	8,425	1.7	0	9%	105	7	12%	1
b14	23,220	10,673	2.5	0	8%	134	12	10%	2
b21	46,502	20,925	5.7	0	7%	522	17	8%	3
aes_core	79,132	45,044	7.6	-	-	-	50	13%	5
b18	121,202	51,977	15.9	-	-	-	40	8%	8
ethernet	280,379	109,519	44.9	-	-	-	108	8%	21
des3_perf	360,624	170,937	51.3	-	-	-	179	8%	24
Average		47,107	14.6	0	7%	161	46	9%	8

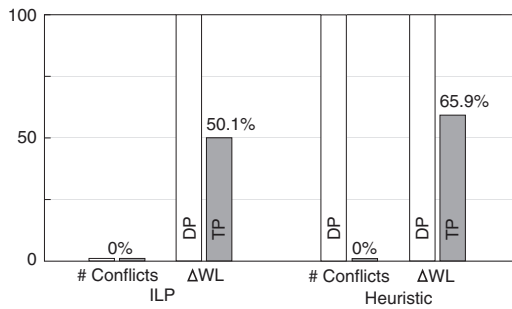


Fig. 10. Average number of coloring conflicts and total wire extensions when cut optimization is performed for triple patterning (TP) and double patterning (DP) using ILP and heuristic algorithm.

We assumed two masks (double patterning or DP) in Table II. With three masks (triple patterning or TP), wider solution space can be explored. It allows coloring conflicts to be resolved more easily and yields less amount of wire extensions. ILP and heuristic algorithm for TP are applied to all test circuits in Table II. Average number of coloring conflicts and total wire extensions are compared in Fig. 10; the results of cut mask optimization for TP are normalized to those for DP. In TP, no coloring conflict remains after ILP and heuristic are applied. Total wire extensions are reduced by 50% with ILP. Heuristic achieves similar result with 35% reduction in total wire extensions.

V. CONCLUSION

We have addressed a cut mask optimization problem for MP-DSAL. It has been formulated as ILP, and fast heuristic algorithm has also been proposed. The heuristic employs divide and conquer strategy. Potential conflict graph is constructed from a given layout and divided into many sub-graphs; each sub-graph is solved optimally. Solution of all sub-graphs are combined together, and local adjustment is performed to remove any remaining coloring conflicts due to ignored edges during graph partition. Our methods have been demonstrated in sub-7nm technology. Experimental results have indicated that our heuristic algorithm achieves a similar result to ILP

with less than 1% coloring conflict and 91% reduction in total wire extensions comparing to the existing approach.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2015R1A2A2A01008037).

REFERENCES

- [1] Y. Badr, A. Torres, and P. Gupta, "Mask assignment and synthesis of DSA-MP hybrid lithography for sub-7nm contacts/vias," in *Proc. Design Automation Conf.*, Jun. 2015, pp. 70:1–70:6.
- [2] J. Ou, B. Yu, and D. Z. Pan, "Concurrent guiding template assignment and redundant via insertion for DSA-MP hybrid lithography," in *Proc. Int. Symp. on Physical Design*, Apr. 2016, pp. 39–46.
- [3] S. Shim, W. Chung, and Y. Shin, "Redundant via insertion for multiple-patterning directed-self-assembly lithography," in *Proc. Design Automation Conf.*, Jun. 2016, pp. 41:1–41:6.
- [4] Z. W. Lin and Y. W. Chang, "Double-patterning aware DSA template guided cut redistribution for advanced 1-D gridded designs," in *Proc. Int. Symp. on Physical Design*, Apr. 2016, pp. 47–54.
- [5] H. Zhang, Y. Du, M. D. F. Wong, and K. Y. Chao, "Mask cost reduction with circuit performance consideration for self-aligned double patterning," in *Proc. Asia South Pacific Design Automation Conf.*, Jan. 2011, pp. 787–792.
- [6] S. Shim, W. Chung, and Y. Shin, "Defect probability of directed self-assembly lithography: fast identification and post-placement optimization," in *Proc. Int. Conf. on Computer Aided Design*, Nov. 2015, pp. 404–409.
- [7] Z. Xiao, C. Lin, and M. D. Wong, "Contact layer decomposition to enable DSA with multi-patterning technique for standard cell based layout," in *Proc. Asia South Pacific Design Automation Conf.*, Jan. 2016, pp. 1–8.
- [8] H. Yi, X. Bao, R. Tiberio, and P. Wong, "Design strategy of small topographical guiding templates for sub-15nm integrated circuits contact hole patterns using block copolymer directed self assembly," in *Proc. SPIE Advanced Lithography*, Mar. 2013, pp. 1–9.
- [9] Z. Xiao, Y. Du, M. D. F. Wong, and H. Zhang, "DSA template mask determination and cut redistribution for advanced 1D gridded design," in *Proc. SPIE Advanced Lithography*, Mar. 2013, pp. 1–8.
- [10] J. Ou, B. Yu, J. R. Gao, D. Z. Pan, M. Preil, and A. Latypov, "Directed self-assembly based cut mask optimization for unidirectional design," in *Proc. Great Lakes Symp. on VLSI*, May 2015, pp. 83–86.
- [11] Z. W. Lin and Y. W. Chang, "Cut redistribution with directed self-assembly templates for advanced 1-D gridded layouts," in *Proc. Asia South Pacific Design Automation Conf.*, Jan. 2016, pp. 89–94.
- [12] Gurobi Optimization, Inc., "Gurobi optimizer reference manual," 2015. [Online]. Available: <http://www.gurobi.com/>.
- [13] "Opencores," <http://www.opencores.org/>.
- [14] "ITC99," <http://www.cerc.utexas.edu/itc99-benchmarks/>.