論文 / 著書情報
Article / Book Information

論題(和文)	
Title(English)	Overlap Area Maximization in Stitch Selection for LELE Double Patterning
著者(和文)	小平 行秀, 高橋 篤司
Authors(English)	Yukihide Kohira, Yoko Takekawa, Chikaaki Kodama, Atsushi Takahashi, Shigeki Nojima, Satoshi Tanaka
出典(和文)	第26回 回路とシステムワークショップ 論文集, , , pp. 466-471
Citation(English)	Proc. the 26th Workshop on Circuits and Systems, , , pp. 466-471
発行日 / Pub. date	2013, 7
URL	http://search.ieice.org/
権利情報 / Copyright	本著作物の著作権は電子情報通信学会に帰属します。 Copyright (c) 2013 Institute of Electronics, Information and Communication Engineers.

Overlap Area Maximization in Stitch Selection for LELE Double Patterning

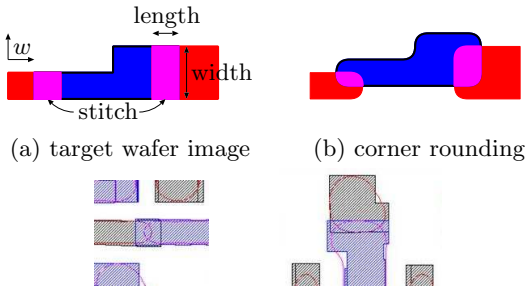
Yukihide KOHIRA† Yoko TAKEKAWA†† Chikaaki KODAMA††
Atsushi TAKAHASHI††† Shigeki NOJIMA†† Satoshi TANAKA††
† The University of Aizu †† Toshiba Corp. ††† Tokyo Institute of Technology

1 Introduction

LELE type double patterning [1] which seems to be the most practical solution for the 22 nm node enables us to fabricate smaller features without using advanced technologies such as extreme ultraviolet (EUV) lithography. In LELE type double patterning, a layout pattern is decomposed and assigned to two masks so that each can be formed on a wafer by an exposure. A layout pattern decomposition method in LELE type double patterning needs to have an ability to obtain a layout pattern decomposition which maximizes the quality of wafer image to achieve higher yield. In layout pattern decomposition for LELE type double patterning, a component in a layout pattern can be partitioned into smaller features and features can be assigned to distinct masks. Adjoining features assigned to different masks are requested to be overlapped to take an overlay error into account. The overlap area between adjoining features which are assigned to different masks is called a stitch. For example, a component is partitioned into three features and assigned to red and blue masks as shown in Fig. 1(a). In the figure, a magenta rectangle represents a stitch.

In practice, the quality of wafer image formed by using stitches is lower than that formed without using stitches. Actual wafer images are degraded due to stitches. For example, wafer images obtained by corner rounding with overlay error and by lithography simulation after OPC (optical proximity correction) with overlay error are shown in Fig. 1(b) and (c), respectively. A layout pattern decomposition without using stitches is preferred. However, stitches are often essential in feasible layout pattern decompositions. Moreover, no feasible layout pattern decomposition often exists even if stitches are used. A layout pattern decomposition is characterized by a set of stitches used in the layout pattern decomposition, called stitch selection. Our problem is to find an optimum stitch selection.

In order to find a better layout pattern decomposition, various methods have been proposed. A pattern segmentation method which identifies stitch-candidates is proposed by Yang et al. in [2]. The segmentation method identifies stitch-candidates effectively but misses some stitch-candidates when a variety of widths of line patterns is large. An efficient polynomial time algorithm to obtain a layout pattern decomposition by using the minimum number of stitches in terms of given stitch-candidates is proposed by Tang and Cho in [3]. However,



(a) target wafer image (b) corner rounding

(c) lithography simulation with OPC

Fig. 1: Stitch and wafer image.

a layout pattern decomposition with minimum number of stitches does not necessarily maximize the quality of wafer image. In order to find a minimum cost stitch selection, a matching based method is proposed by Xu and Chu in [4]. The method utilizes the planarity of a constraint graph, and detects faces of the constraint graph which are needed to be broken by stitches to obtain a feasible layout pattern decomposition. These faces are matched by using a minimum cost stitch selection and are broken. However, how a stitch is inserted is not well-defined in the method especially when a variety of widths of line patterns is large.

In this paper, we propose an efficient algorithm which obtains a minimum cost stitch selection. First, stitch-candidates which help not to degrade the quality of wafer image are extracted. The extracted stitch-candidates are independent of other stitch-candidates. Then, a minimum cost stitch selection which derives a feasible layout pattern decomposition is obtained from the stitch-candidates. By using the cost of a stitch which reflects the quality degradation on wafer image appropriately without degrading the stability and efficiency of computation, a layout pattern decomposition which has the maximum wafer quality is obtained efficiently from the minimum cost stitch selection.

Our proposed algorithm is matching based. The algorithm flow is similar to the algorithm proposed in [4]. In [4], stitch-candidates are assumed to be independent of other stitch-candidates, and the conflict graph where each vertex corresponds to a component is assumed to be planar. While, in our proposed algorithm, one stitch-candidate is defined in each interval where a stitch can be inserted without violating the design rule. A stitch-candidate may cross each other when the width of a line pattern is large, but is independent of other stitch-

candidates, that is, no design rule violation occurs in any stitch selection obtained from the defined stitch-candidates. A component in a layout pattern is decomposed into primitive shapes by using stitch-candidates. The conflict graph where each vertex corresponds to a primitive shape is defined in our proposed algorithm. Our algorithm obtains the minimum cost stitch selection when the conflict graph defined is planar. Our algorithm does not handle a problem instance where the conflict graph is not planar, though most of problem instances have planar constraint graph.

The quality degradation caused by stitch is mitigated by setting the minimum length or area of stitch. Since the stitch length must be large enough to tolerate the overlay error and etc., the minimum stitch length is given. A stitch-candidate is defined when its length can be larger than or equal to the minimum length without violating the design rule. As confirmed in Fig. 1(b) and (c), a narrow short stitch is more fragile than a wide long stitch. The quality degradation decreases by increasing the stitch length and width, but the quality remains the same when its length and width are large enough even if the stitch becomes longer and wider. So, upper bounds of stitch length and width which are used in cost evaluation are also given as user-defined thresholds. In our proposed stitch cost, the area of a stitch is taken into account to reflect the quality degradation by lithographic process.

2 Preliminaries

In the layout area, components in one layer are given as input. A component is a maximal connected region in the layout area. In layout pattern decomposition, a component is partitioned into smaller features with overlap, and a feature is assigned to one of two masks. The overlap area between adjoining features which are assigned to different masks is called a *stitch*.

A line segment between two adjacent corners of a boundary of a feature or component is called a *seg*. A seg is either horizontal or vertical and distinct seg neither touch nor cross each other. A stitch is rectangular region which disconnects a component. In order to specify the position of a stitch, the centerline of the stitch is used. The centerline of a stitch is a horizontal or vertical slice-line of a component. The orthogonal stitches may overlap each other.

In this paper, Manhattan distance is used to define our assumed design rule. Let $d_A(p,q)$ and $d_B(p,q)$ be the distances between points p and q in the layout area and in components, respectively. Note that $d_B(p,q)$ is the minimum length of routes connecting p and q in components. If there is no route between them, the distance is defined infinite. Apparently, $d_A(p,q) \leq d_B(p,q)$. Let $d_A(P,Q)$ and $d_B(P,Q)$ be the distances between objects P and Q in the layout area and in components, respectively. The minimum length of the horizontal or vertical slice-line of a component (feature) is called a *width* of a component (feature).

The design rule used is described by using four constraints, called *width*, *spacing*, *double-patterning-spacing* (*dp-spacing*), and *stitch-length*. The width constraint is violated if and only if there is a feature or component whose length is less than w_w. The spacing constraint is violated if and only if there is a pair of points p and q in components such that $d_A(p,q) < w_s$ and $d_A(p,q) < d_B(p,q)$. The dp-spacing constraint is violated if and only if there is a pair of points p and q in components in a mask such that $d_A(p,q) < w_d$ and $d_A(p,q) < d_B(p,q)$. The stitch-length constraint is violated if and only if the length of a stitch is less than w_t. In Fig. 2, when $w_d = 3w$, points p and q satisfy the dp-spacing constraint, but points q and r do not. Although different threshold values can be used in our design rule as used in practice, a simplest design rule is used to explain the concept of our method.

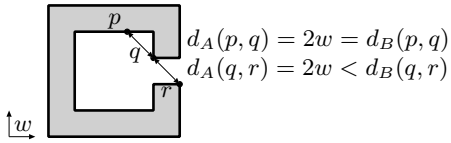


Fig. 2: Dp-spacing is satisfied between p and q but not between q and r ($w_d = 3w$).

In double patterning technology, the assumption that parallel line patterns of $w_w + w_s$ pitch can be fabricated by LELE but cannot by a single mask is natural. If $w_s \geq w_d$ then a single mask is enough to realize the layout pattern. If $w_d > w_w + 2w_s$ then the layout pattern cannot be realized by LELE. Therefore, we assume that $w_s < w_d \leq w_w + 2w_s$ in the following. Also, we assume that $w_t \leq w_w$ though it is not mandatory but it is natural and the explanation is easy.

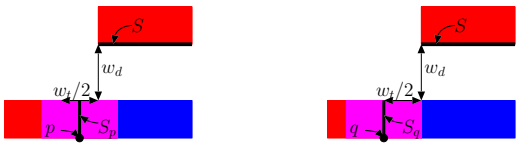
In the following, we focus on a layout pattern which satisfies width and spacing constraints. A layout pattern decomposition, a decomposition hereinafter, is said to be *legal* if width, spacing, dp-spacing, and stitch-length constraints are satisfied.

Stitch-candidates are defined by partitioning a component into features. The details are discussed in section 3. A stitch-candidate becomes a stitch in a decomposition if both sides of it are assigned to different masks in the decomposition. A set of stitches which defines a decomposition is called a *stitch selection*. A stitch selection is said to be *legal* if a legal decomposition is obtained by it. A positive *cost* is set to each stitch-candidate which reflects the impact on the quality of wafer image. The cost of a stitch selection is the sum of costs of stitches in the selection. Our problem is to find a minimum cost legal stitch selection.

3 Stitch-Candidate

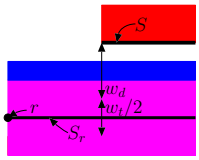
In this section, stitch-candidates which are generated in our proposed algorithm are defined. Our proposed algorithm excludes stitch-candidates that violate the design rule. First, two types of locations where stitch-candidate is not allowed are discussed. Then, the cost of stitch used in this paper is defined.

First type is caused by narrow spacing in a layout pattern. If the distance between boundaries of com-



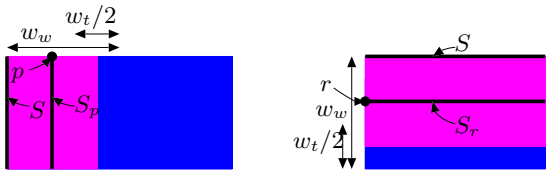
(a) Too close to the adjacent component

(b) Valid stitch-candidate



(c) Invalid stitch-candidate on major axis of thick line

Fig. 3: Points p and r are on the alley wall, but point q is not.



(a) Orthogonal to the major axis

(b) Parallel to the major axis

Fig. 4: Points p and r are on the corner wall.

ponents is less than w_d, then no stitch which cuts these boundaries can be used without violating the dp-spacing constraint. In order to prevent from generating stitch-candidates at narrow spacing location, *alley walls* are defined on the boundary of a component. Formally, alley walls are defined as the set of points p on boundaries of components such that the slice-line S_p of a component which ends at p has a seg S such that $d_A(S_p, S) < w_d + w_t/2$ and $d_A(S_p, S) < d_B(S_p, S)$. For example, point p in Fig. 3(a) and point r in Fig. 3(c) are contained in alley wall, but point q in Fig. 3(b) is not.

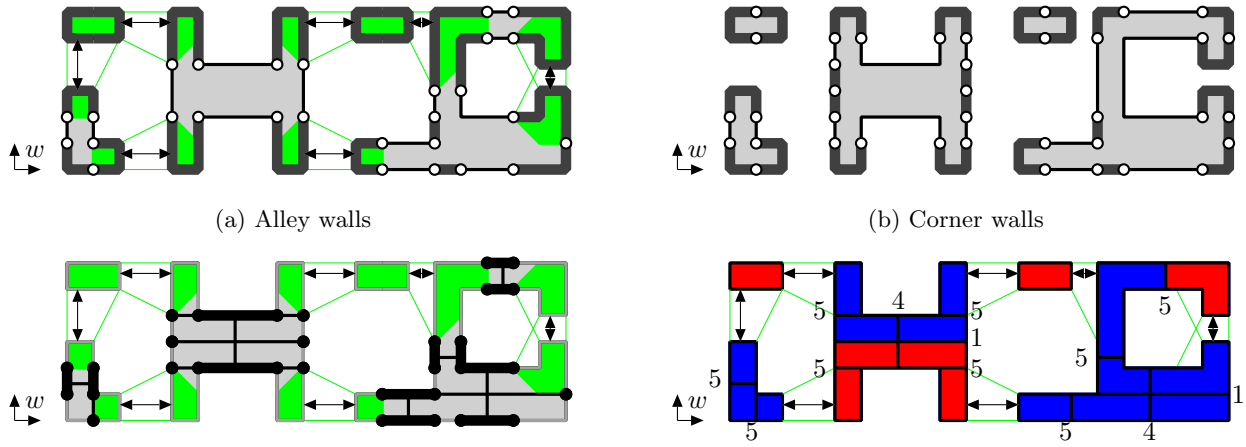
Second type is caused by corner of a component such as line ends. If a stitch is near an end of line pattern, then a short feature which violates the minimum width constraint is generated. In order to satisfy the minimum width constraint and the minimum stitch length constraint, the centerline of a stitch cannot be located near an end of line pattern. In order to prevent from generating stitch-candidates which generate a short feature, *corner walls* are defined on the boundary of a component. Formally, corner walls are defined as the set of points p on boundaries of components such that the slice-line S_p of a component which ends at p has a seg S of the same component which is parallel to S_p and $d_A(S_p, S) < w_w - w_t/2$. For example, points p and r in Fig. 4 are contained in corner wall.

Corner wall and alley wall are simply called *wall*. Obviously, when the centerline of a stitch whose length is minimum cuts a wall, the stitch is not contained in any legal stitch selection. On the other hand, a stitch which does not cut a wall can be used in a legal stitch selection without violating design rule when its length is minimum. In our algorithm, an interval which consists of maximal parallel boundaries defined by excluding walls is used to define a stitch-candidate. A slice-line which connects the pair of parallel outer boundaries of an interval is used as the centerline of a stitch-candidate. By defining one stitch-candidate for each interval, stitch-candidates are independent of other stitch-candidates when these lengths are minimum. Although the position of the centerline of a stitch-candidate is arbitrary within each interval, the centerline is set to the center of the interval so that the length of the stitch can be increased as much as possible if necessary without affecting other stitch candidates. A *feature* is defined as a maximum connected region of a component obtained by cutting the component by the centerlines of stitches.

For example, the alley walls and corner walls are shown in Fig. 5(a) and (b), respectively. They are represented by thick gray lines. In Fig. 5(a), a green region represents area where a stitch cannot pass without violating the dp-constraint. In Fig. 5(c), the outer boundaries of intervals where a stitch can be inserted without violating the design rule are represented by thick black lines. An end of each interval is closed. A line connecting outer boundaries of an interval represents the centerline of a stitch-candidate. The numbers of features and stitch-candidates are 20 and 13, respectively. In Fig. 5(d), the cost of each stitch-candidate just for reference and an example of decomposition into red and blue using 2 stitches are shown. A stitch-candidate is a straight line segment inside a component connecting the boundary of the component. The number of stitches in this decomposition is minimum and the cost is 6.

The segmentation method in [2] decomposes components by using projection method to separate green and gray regions shown in Fig. 5(a), and then makes each part rectangular. The centerlines of stitch-candidates do not intersect each other and a wide stitch would not be found. Also, the length flexibility is low though a post processing would increase the length of a stitch. An example of segmentation using the method in [2] with decomposition is shown in Fig. 6. The number of stitches in the decomposition is 2 and the cost of the stitch selection is 10. The number of stitches obtained by the method in [3] is not minimum in general if the segmentation method in [2] is used. Such an example is obtained from the layout pattern shown in Fig. 5(d) by removing the red feature in the right component.

The cost of stitch used in this paper is defined as follows. Let $l(s)$ and $w(s)$ be the length and width of a stitch (or a stitch-candidate) s, respectively. In order to take the maximum possible stitch length in evaluation, $l(s)$ is defined as the sum of w_t and the length of the interval at which s is defined. The effective length $l'(s)$ and width $w'(s)$ of s are defined as $l'(s) = \min\{l(s), L_{\max}\}$ and $w'(s) = \min\{w(s), W_{\max}\}$, respectively, where $L_{\max}$ and $W_{\max}$ are user-defined thresholds. Let $c(s)$ be the cost of a stitch (or a stitch-candidate) s. In this paper,



(a) Alley walls (b) Corner walls

(c) Intervals and stitch-candidates (d) Features, stitch cost, and decomposition of cost 6

Fig. 5: Features, stitch-candidates, and decomposition with 2 stitches ($w_w = w_s = w$, $w_d = 3w$, $w_t = 0$).

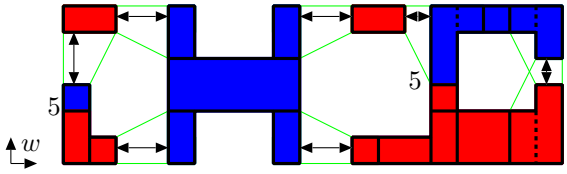


Fig. 6: Decomposition of cost 10 according to the segmentation method in [2].

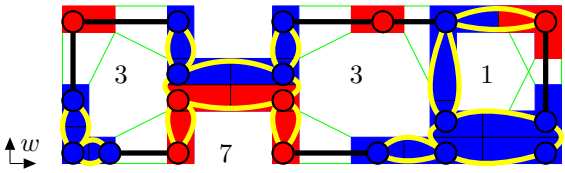


Fig. 7: Relation graph and decomposition with 2 stitches.

the cost is defined as

$$c(s) = L_{\max} W_{\max} - l'(s)w'(s) + C$$

where C is a big constant. In this cost function, the area of s which is defined by the area of s which is given as $l'(s)w'(s)$ is taken into account. The cost is smaller if the length (width) is larger when the length (width) is at most $L_{\max}$ ($W_{\max}$). If C is large enough, then the minimum number of stitches is guaranteed while the sum of areas is maximized.

4 Legal Stitch Selection

In this section, the legality of stitch selection is discussed by defining conflict graph and relation graph. In order to obtain a legal stitch selection, the *conflict graph* which represents the relation between features is defined. A vertex in the conflict graph corresponds to a feature. An edge in the conflict graph is inserted between vertices corresponding to features b_i and b_j if and only if there is a pair of points $p \in b_i$ and $q \in b_j$ such that $d_A(p,q) < w_d + w_t/2$ and $d_A(p,q) < d_B(p,q)$. Note that an edge in the conflict graph is added between vertices which correspond to features with narrow spacing which cannot be assigned to the same mask simultaneously. A layout pattern has a legal stitch selection if and only if the conflict graph of the layout pattern is bipartite.

A bipartition is not unique in general. Our problem is to obtain a minimum cost stitch selection which corresponds to a bipartition. The stitch selection without stitches is called the *empty stitch selection*. If the empty stitch selection is legal, then it is the minimum cost legal stitch selection. Although it is easy to check whether the empty stitch selection is legal, it is not trivial for an arbitrary stitch selection. In order to check whether a stitch selection is legal, the *relation graph* is defined from the conflict graph of the layout pattern. In the relation graph, an edge between vertices which correspond to features adjoining at the boundary of a component and which are cut by a stitch-candidate not contained in the stitch selection is added. In the relation graph, edges corresponding to edges in the conflict graph are called *conflict-edges*, and the others are called *boundary-edges*. A cycle in a relation graph is said to be *illegal* if the number of conflict-edges in the cycle is odd. A stitch selection of a layout pattern is legal if and only if the relation graph of the stitch selection contains no illegal cycle, that is, the number of conflict-edges in every cycle is even.

A minimum cost legal stitch selection is a stitch selection such that the number of conflict-edges in every cycle in the relation graph is even and that the total cost is minimum. In Fig. 7, the relation graph of the empty-stitch selection of the layout pattern shown in Fig. 5 is shown. Black and yellow edges represent conflict-edges and boundary-edges, respectively. This relation graph contains illegal cycles, but they are broken if 4 boundary-edges which corresponds to 2 stitches in the shown decomposition are removed.

5 Odd Face of Layout Pattern

The planarity of relation graph helps to find a minimum cost legal stitch selection, though the conflict graphs defined for double patterning are not planar in general as mentioned in [3]. A relation graph is not

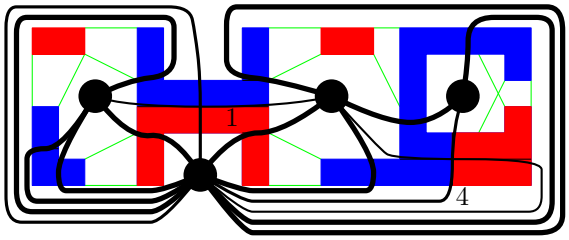


Fig. 8: Face graph and the minimum cost legal stitch selection with decomposition of cost 5.

necessarily planar as well. However, when $w_s < w_d \leq w_w + 2w_s$ and $w_t \leq w_w$, there is a natural embedding of a relation graph onto plane where the embedding of each edge corresponds to a shortest path that does not pass the other components. We focus on a layout pattern where edges do not cross in such embedding, and a plane embedding of a relation graph is used in the following.

Let G_r be a plane embedding of a relation graph. A face of G_r that does not have conflict-edge corresponds to features of the layout pattern. In the following, a face of G_r with at least one conflict-edge is focused. The *parity* of a face of G_r is defined as the parity of the number of conflict-edges in the edge set, counting bridges twice. The number of conflict-edges in each face is shown in Fig. 7. The number of odd faces of G_r is even. Note that if G_r contains no odd face, then the number of conflict-edges in every cycle of G_r is even and the corresponding stitch selection is legal. When a stitch is added to the stitch selection, the boundary-edges which correspond to the stitch are removed from G_r, and two faces of G_r are merged into one. A minimum cost legal stitch selection is obtained from the empty stitch selection by adding stitches with minimum cost so that all the odd faces of G_r are eliminated.

In order to characterize a minimum cost legal stitch selection, the *face graph* which represents the relation between faces is defined. A node in the face graph corresponds to a face of G_r with at least one conflict-edge where G_r is a plane embedding of the relation graph of the empty stitch selection. An edge in the face graph is inserted between vertices corresponding to faces connected by a stitch-candidate. The weight of an edge is positive and corresponds to the cost of the corresponding stitch-candidate. The multiple edges between two nodes may be merged if preferred. A minimum cost legal stitch selection corresponds to a set of paths connecting odd faces which eliminates all the odd faces whose total cost is minimum. In order to find a minimum cost legal stitch selection, the *cost graph* of a layout pattern is defined. A node in the cost graph corresponds to an odd face of the relation graph of the empty stitch selection. An edge in the cost graph is inserted between vertices if and only if there is a path connecting them in the face graph. The cost of an edge is the weight of a shortest path connecting them in the face graph. If the conflict graph is bipartite, then a perfect matching exists in the cost graph. A stitch is contained at most once in a minimum-cost perfect matching in the cost graph. A stitch selection is a minimum cost legal stitch selection if and only if it corresponds to a minimum-cost perfect matching of the cost graph.

For example, the face graph of the layout pattern shown in Fig. 5 is shown in Fig. 8. The number of nodes in the face graph is 4. All faces that have conflict-edges are odd. The weight of an edge corresponds to the cost of the corresponding stitch. The cost of a minimum-cost perfect matching in the cost graph is 5. The decomposition corresponding to the minimum cost stitch selection is also shown.

6 Algorithm

According to the analysis described in the previous sections, an algorithm which finds a minimum cost legal stitch selection is proposed as follows.

1. Extract all the stitch candidates in the layout pattern. Construct the conflict graph and the relation graph of the empty stitch selection, and specify faces.
2. If the conflict graph is not bipartite, then report odd faces of the relation graph in order to help to modify the layout pattern. If necessary, delete conflict edges to break the corresponding odd faces, and continue.
3. If the number of odd faces is zero, then output the empty stitch selection with decomposition, and terminate.
4. Construct the face graph from the relation graph of the empty stitch selection, and then construct the cost graph from the face graph by finding shortest paths between odd faces.
5. Find a minimum-cost perfect matching in the cost graph, and output the corresponding stitch selection with decomposition.

The conflict graph and the relation graph can be constructed in $O(n^2)$ time by a naive implementation, where n is the number of segs in the layout pattern. However, we believe that the time complexity of constructing these graphs is almost linear after sorting the segs in practice. The face graph is not necessarily planar but is constructed in $O(n)$ since the number of conflict-edges and the number of stitch-candidates are $O(n)$. The cost graph is constructed in $(n_p^2 n_o)$ where n_p and n_o are the number of faces and the number of odd faces, respectively. A minimum-cost perfect matching of a graph can be obtained by a straightforward implementation of Edmonds's algorithm in $O(n_o^2 m_o)$ time where n_o and m_o are the number of nodes and the number of edges in the cost graph. In our implementation, the minimum-cost perfect matching is obtained by a practical implementation due to Kolmogorov [5].

7 Experiments

We implemented the proposed stitch selection method in C++ language, and the method is executed on a Linux machine with 6 GB memory by using single Intel core i7-940 of 2.93 GHz.

First, our method is evaluated by comparing with state-of-the-art methods in terms of the number of

Table 1: ISCAS benchmarks. ([2,3]: 3.0 GHz, 4 GB memory, Ours: 2.93 GHz, 6 GB memory)

Benchmark								ILP [2]	Tang [3]		Ours	
name	#comp	#seg	#ce	#fc	#ofc	#sc	#st	tot(s)	tot(s)	sol(s)	tot(s)	sol(s)
c432	850	4918	540	518	2	1028	1	0.63	0.23	0.03	0.07	0.00
c499	1491	9518	1489	1002	100	2067	50	100.0	0.47	0.03	0.10	0.00
c880	1872	10666	1422	984	344	2472	198	4525.6	0.44	0.03	0.12	0.01
c1355	2656	15246	1514	1514	164	3290	114	702.4	0.49	0.02	0.17	0.00
c1908	4191	24370	3141	2416	418	5267	371	37019.8	1.10	0.04	0.39	0.01
c2670	6371	37564	5802	3543	1350	8769	947	>24Hr	2.00	0.11	0.83	0.13
c3540	8188	47244	6897	4501	1622	10914	1034	>24Hr	2.64	0.14	1.20	0.21
c5315	11498	68476	10097	6451	2464	16464	1545	>24Hr	4.55	0.24	2.27	0.48
c6288	11605	64762	5602	6515	512	15354	256	>24Hr	3.23	0.25	1.40	0.03
c7552	17167	99526	14027	9376	3046	22590	2058	>24Hr	8.19	0.32	4.14	0.79

stitches. ISCAS benchmarks used in [2,3] are used which are reproduced from the information given by authors of [2,3] and from figures in [2], though we could not obtain the same data. In this experiment, the cost $c(s)$ of stitch s is set to 1, and we followed the parameters as in [2,3] which are $w_d = 54$ nm and $w_t = 20$ nm. The number of stitches by our method is same as the other methods for each layout pattern. In Table 1, results are summarized.

In tables, #comp, #seg, #ce, #fc, #ofc, #sc, and #st represent the number of components (patterns), the number of segs, the number of conflict-edges, the number of faces, the number of odd faces, the number of stitch-candidates, and the number of stitches used that is equal to the minimum number of stitches in a legal decomposition, respectively. "tot" is the total execution time and "sol" is the time excluding the time to construct the relation graph and to output the result. The results of ILP and Tang in Table 1 are directly copied from [3]. We believe that the impact on computation time caused by the difference of experimental setting is not big. Our implementation based on the proposed approach is two times faster than [3] in total time. In our current implementation, no speedup techniques are employed to handle larger layout patterns. Though solver time for larger layout patterns is larger than [3], speedup techniques discussed as in [3,4] will reduce the computation time much for larger layout patterns.

Next, the cost of stitch selection is evaluated. A minimum cost legal stitch selection obtained by our method is compared with other stitch selections. A layout pattern is generated based on PDKv1_3_v2010_12 in Nangate FreePDK45 Library [6]. Among several layers in the library, Metal1 is used for evaluation. We assume the layout at 40 nm generation and apply appropriate decomposition rules and lithography conditions. The parameters are set as follows: $w_w = w_s = 65$ nm, $w_d = 70$ nm, $w_t = 20$ nm, $L_{\max} = 80$ nm, $W_{\max} = 400$ nm, $C = 10L_{\max} \cdot W_{\max}$ nm^2. In this setting, the minimum number of stitches is not guaranteed, but the number of stitches is minimum in experiments.

In Tables 2 and 3, results are summarized. In these tables, "Min-stitch" is obtained from a stitch-selection with the minimum number of stitches. "Min-stitch l-extend" is obtained from the stitch-selection "Min-stitch" with the minimum number of stitches by extending the length of each stitch as much as possible. "Max-length" is obtained by using $(L_{\max} - l'(s)) + 10L_{\max}$ as the cost function. "Max-area" is the result of our method. The statistics of the layout pattern and the result of our method are shown in Table 2. In Table 3, "len", "width", and "area" represent the sum of stitch lengths $l'(s)$, the sum of stitch widths $w'(s)$, and the sum of stitch areas $l'(s)w'(s)$, respectively. Experimental results show that our method obtains the optimum decompositions according to the given cost functions.

Table 2: Layout pattern by Open Cell Library.

name	#comp	#seg	#ce	#sc	#st	tot(s)	sol(s)
55x55	12626	93580	13907	36231	776	3.618	0.169

Table 3: Evaluation of stitch selections.

condition	#st	len (μm)	width (μm)	area (μm^2)
Min-stitch	776	15.520	62.755	1.255
Min-stitch l-extend	776	54.785	62.755	4.359
Max-length	776	56.184	61.340	4.402
Max-area (Ours)	776	56.169	68.315	4.955

8 Conclusions

In this paper, we propose an efficient algorithm which finds a set of stitches which gives a feasible layout decomposition for litho-etch-litho-etch (LELE) type double patterning with the minimum cost. The correctness and efficiency of the proposed algorithm is theoretically guaranteed, and the validity of our implementation is confirmed by experiments. The quality of stitch selection in terms of the lithograph compliance depends on the cost assigned to each stitch-candidate. The definition of cost of stitches which maximizes the total yield by taking mask density balance, stitch direction, and etc. into account is in our future works.

References

[1] G.E. Bailey, A. Tritchkov, J.-W. Park, L. Hong, V. Wiaux, E. Hendrickx, S. Verhaegen, P. Xie, and J. Versluijs, "Double pattern EDA solutions for 32nm HP and beyond," Proc. SPIE, vol.6521, p.65211K, 2007.

[2] J.-S. Yang, K. Lu, M. Cho, K. Yuan, and D.Z. Pan, "A new graph-theoretic, multi-objective layout decomposition framework for double patterning lithography," Proc. ASP-DAC, pp.637–644, 2010.

[3] X. Tang and M. Cho, "Optimal layout decomposition for double patterning technology," Proc. ICCAD, pp.9–13, 2011.

[4] Y. Xu and C. Chu, "A matching based decomposer for double patterning lithography," Proc. ISPD, pp.121–126, 2010.

[5] V. Kolmogorov, "Blossom V: a new implementation of a minimum cost perfect matching algorithm," Mathematical Programming Computation, vol.1, pp.43–67, July 2009.

[6] "Nangate Open Cell Library," `http://www.si2.org/openeda.si2.org/projects/nangatelib`.