論文 / 著書情報
Article / Book Information

Title	Rip-up and Reroute based Routing Algorithm for Self-Aligned Double Patterning
Authors	Takeshi Ihara, Atsushi Takahashi, Chikaaki Kodama
Citation	Proc. the 19th Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI 2015), Vol. , No. , pp. 83-88
Pub. date	2015, 3

Rip-up and Reroute based Routing Algorithm for Self-Aligned Double Patterning

Takeshi Ihara and Atsushi Takahashi

Department of Communications and Computer Engineering
Tokyo Institute of Technology
Tokyo 152-8850
Email: {ihara,atsushi}@eda.ce.titech.ac.jp

Chikaaki Kodama

Toshiba Corporation
Semiconductor & Storage Products Company
Yokohama, Kanagawa 247-8585
Email: chikaaki1.kodama@toshiba.co.jp

Abstract— Self-Aligned Double Patterning (SADP) is an important manufacturing technique for sub 20 nm technology node. Although various routing algorithms for SADP have been proposed, the quality of a routing pattern generated by them is not good enough. In this paper, a rip-up and reroute based routing algorithm for SADP is proposed to obtain a more reliable routing pattern efficiently. In SADP, a cut pattern which is introduced in pattern mask reduces the extra mask cost, but a cut pattern itself potentially degrades the reliability of image on a wafer. The proposed algorithm generates a routing pattern that needs less cut patterns by preventing a net from touching to other routes many times. In experiments, it is shown that the number of cut patterns is reduced by our proposed algorithm with less length overhead.

I. Introduction

In VLSI manufacturing, the difficulties in forming pattern on wafer are significantly increased under Arf immersion lithography, and various kinds of multiple patterning techniques are being developed to alleviate the difficulties. Among multiple patterning techniques, *Self-Aligned Double Patterning* (SADP) [1,2] is considered as an important manufacturing technique for sub 20 nm technology node.

In SADP, single litho-etch process is used to form a pattern on wafer, and a fine pitch is achieved by slimming and sidewall spacer process. The quality of a wafer image obtained by SADP is expected to be better than that obtained by *Litho-etch-litho-etch* (LELE) [3]. In LELE, a target layout is decomposed into two pattern masks and the superposition of them realizes the target layout. Various methods that find a better layout pattern decomposition have been proposed [4,5]. The overlay error caused between two pattern masks in LELE is expected to be larger than the error caused by slimming and sidewall spacer process in SADP. Although a better quality is expected to be achieved by SADP, a target layout in SADP must satisfy tighter constraints than LELE. It is not easy to obtain a pattern that can be manufactured by SADP. A stitch which is allowed in LELE is not allowed in SADP. Also, all routing grids which are not used in target pattern must be filled by dummy patterns in SADP. The mask for SADP is not intuitive since it contains a subset of patterns as well as a subset of dummy patterns.

Several design methods that generate two-dimensional pattern for SADP have been proposed so far [6–11]. Mirsaeedi et al. proposed a grid routing algorithm for SADP [6], but how to avoid conflicts during routing process was not well discussed. Even though methods using trim mask or cut mask after SADP process were proposed [7–9], the routing design is complex and the cost of mask and the effect of overlay error must be considered.

In order to realize a target pattern by SADP without trim and cut mask, a special mask pattern, called *cut-pattern*, in pattern mask was introduced in [10]. The manufacturability by SADP without using additional mask is achieved by introducing cut-patterns in pattern mask. In [10], a routing algorithm that generates routing patterns by using a partially pre-colored two-color base grid was proposed. A routing pattern that can be manufactured by SADP is obtained without complex constraints in routing by using the two-color base grid. However, the routing algorithm is proposed in [10] has a limitation on pin location of a net. The limitation can be eliminated by adopting the routing approach proposed in [11]. The approach proposed in [11] recolors several pre-colored grids to eliminate the limitation, but recoloring is kept as small as possible to maintain the advantages which are obtained by the regularity of the two-color base grid.

Cut-pattern in pattern mask reduces the necessity of extra mask in SADP. However, the reliability of a wafer image around a cut-pattern is expected to be inferior to other area. Cut-pattern might cause potential defects in wafer image. Therefore, a routing pattern that requires less cut-patterns is better. However, the reliability degradation which might be caused by cut-patterns is not taken into account in routing pattern generation in [10] and [11].

In this paper, a routing algorithm that can be manufactured by SADP with less cut-patterns is proposed under the design approach proposed in [11]. The proposed routing algorithm generates routing patterns using A^* algorithm with rip-up and reroute technique. The proposed

routing algorithm reduces the number of cut-patterns by introducing a cost that controls the length of parallel segments. Experiments show that the number of cut patterns is reduced by introducing a cost that prevents a net from touching to other routes many times.

II. Preliminaries

A. *Manufacturing process of Self-Aligned Double Patterning*

Self-Aligned Double Patterning (SADP) has two main processes, *Spacer-Is-Dielectric* (SID) and *Spacer-Is-Metal* (SIM) [12,13]. In this paper, we focus on SID-type SADP as focused on [10,11] since the design flexibility and overlay controllability are larger than SIM-type SADP as mentioned in [14]. In the following, we simply call SID-type SADP as SADP.

The manufacturing process of SADP is briefly explained in the following, which is also illustrated in Fig. 1. (a) a pattern with double pitch of a target pattern is formed on a wafer by lithography process of a single exposure. The pattern formed by this exposure is called *mandrel*. (b) the width of each mandrel pattern is halved by slimming process. (c) the sidewall spacer is formed on both sides of each mandrel pattern by depositing masking material. (d) the sidewall spacer is etched and mandrel is exposed. (e) mandrel is removed. (f) the substrate is etched with the sidewall spacer as a mask and the spacer is removed. (g) the final patterns of target pitch is formed by filling the etched area by conductive material.

Note that the line pitch obtained by SADP is a half of the line pitch of mandrel which is formed by optical lithography. A pattern component in a final pattern formed by SADP is categorized into two types. One, called *primary pattern*, corresponds to a mandrel pattern, and the other, called *secondary pattern*, corresponds to a final pattern formed between mandrel. Since no stitch is allowed in SADP, the connection of each net is realized by either type of a pattern component. Therefore, the flexibility of pattern that can be formed by SADP is lower than LELE.

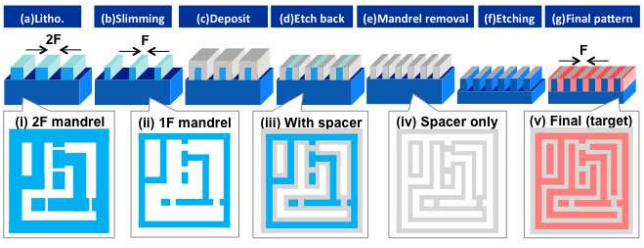
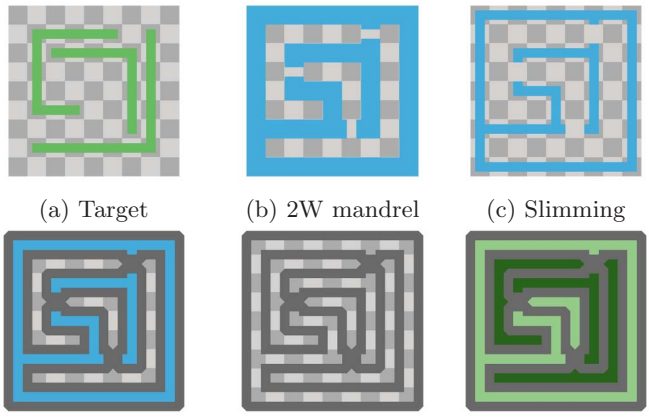


Fig. 1. Self-Aligned Double Patterning (SADP) manufacturing process in [10]



(a) Target (b) 2W mandrel (c) Slimming

(d) Sidewall Deposit (e) Mandrel Removal (f) Final Pattern

Fig. 2. Grid base pattern representation

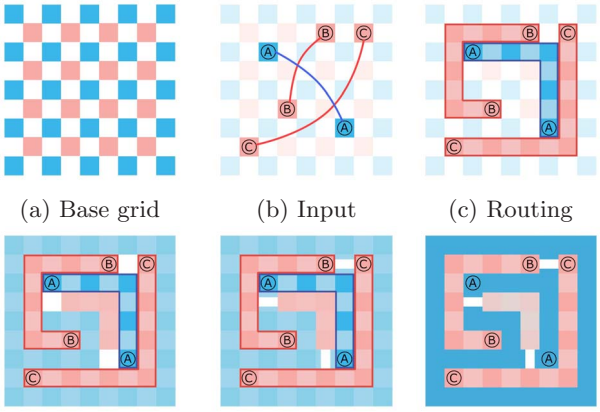
B. *Grid base pattern representation*

A pattern formed by SADP is represented by using a uniform grid structure where the pitch of the grid is equal to the pitch of final pattern (see Fig. 2). A pin of a net is assigned to a grid point. Basically, type of a grid point is either primary or secondary. Adjacent grid points are connected in the final patterns if the types of them are the same. The connections of a net is realized in the final pattern if all pins of the net are connected and no pin of the others is connected. When a target pattern that realizes the connections of nets are given, the type of each pattern component in the target pattern must be decided not to create any undesired connections. In addition, all grid points must be filled by either primary or secondary unless trim mask, cut mask, and special pattern are used.

C. *Cut-pattern in pattern mask*

A grid point not used to realize the connection of a net is called a *dummy grid point*. A pattern component formed at a dummy grid point must not create any undesired connections. However an undesired connection is often created even if either type is assigned to a dummy grid point. In order to create no undesired connections at dummy grid points without using trim and cut masks, a special mask pattern called *cut-pattern* which was introduced in [10] is assigned to a dummy grid point if necessary. The cut-pattern enable us to obtain a feasible pattern by SADP without using trim and cut masks. For example, two cut-patterns are used in Fig. 2.

Cut-pattern is required when routes of different nets are mutually adjacent on the grid. No cut-pattern is required if no route touches to the other routes. A routing pattern without touching to other routes will be obtained if the congestion of routing area is small enough. However, it is an impractical assumption in cases where SADP process is required. In these cases, the existence of cut-pattern is essential to realize connection requirements. Even if the



(a) Base grid (b) Input (c) Routing

(d) Dummy pattern (e) Cut pattern (f) Mandrel

Fig. 3. Routing on partially pre-colored two-color base grid

existence of cut-pattern is inevitable, the occurrence of cut-patterns will be suppressed by controlling route of a net appropriately.

III. Base Grid Based SADP Pattern Generation

A. *Partially pre-colored two-color base grid*

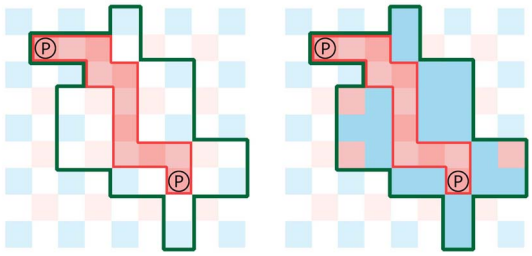
A partially pre-colored two-color base grid was introduced in [10] to generate a routing pattern that can be realized by SADP. In the base grid, the color of a grid point is either blue, red, or white, and is arranged as shown in Fig. 3(a).

In the routing method used in [10], the type of a blue grid point is decided to blue and the type of a red grid point is decided to red. The connections of nets are realized by deciding each white grid point to either blue or red. Then, the type of each dummy grid point is decided to either blue, red, or blank so that no undesired connections are created. The cut-pattern is assigned to blank grids. The routing method assumes that all the pin of a net are on the same color grid points. The pattern generation procedure is illustrated in Fig. 3.

B. *Double-colored net*

In general, there is a net that contains pins on grids of both colors on the partially pre-colored two-color base grid. A net is said to be *double-colored* if it contains both of pins on blue and red grid points, and is said to be *single-colored* otherwise. A single-colored net is either blue or red. A blue (red) net contains no red (blue) pins.

The routing method used in [10] generates the route of a single-colored nets, but does not handle double-colored nets. While the design approach proposed in [11] handles both single-colored nets and double-colored nets. In the design approach proposed in [11], the route of a double-colored nets is generated on the partially pre-colored two-color base grid by allowing recoloring of the base grid.



(a) Influenced Area (b) Pattern

Fig. 4. Influenced Area

A pin of a net is said to be blue, red, and white if it is on blue, red, and white grid point, respectively. The type of a pin is decided to either blue or red. The type of all pins of a net must be decided to the same to realize the connection of the net by SADP. A pin placed on a white grid can be handled by deciding the color either blue or red at the expense of a redundant extension of the connection. Recoloring of the base grid is inevitable to realize the connection of a double-colored net.

If the route of other net uses a grid point near a recolored grid point, the final pattern might not be manufactured correctly by SADP. An example of influenced area of the route of double-colored net is shown in Fig. 4(a).

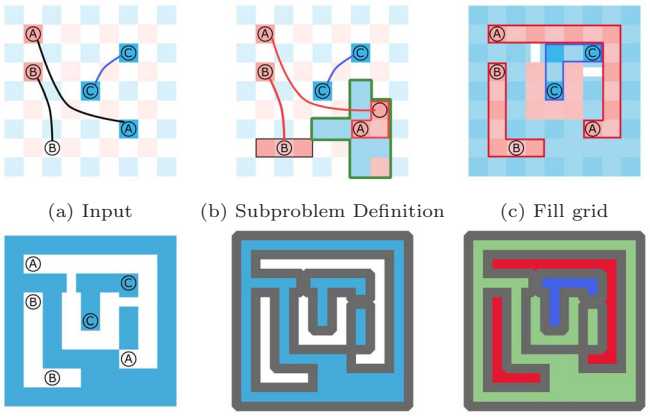
In the design approach proposed in [11], the design inside the influenced area is decided as shown in Fig. 4(b). Grid points except the route of a double-colored net are used as dummy and are not used for the routing of other nets. The type of dummy is decided to the appropriate color so as not to be adjacent to same color route. A dummy pattern that surrounds the route limits the effect of recoloring. In order to minimize the number of cut-patterns around the boundary of the influenced area, the final color assignment to a grid inside a influenced area is decided after the routing pattern of single-colored nets is fixed.

IV. SADP-friendly routing algorithm

A. *Overview*

An overview of the proposed routing algorithm that follows the design approach proposed in [11] is explained with an illustrative example shown in Fig. 5.

First, the type of a white pin is decided so as not to increase the double-colored nets. Second, each double-colored net is divided into a short double-colored subnet and a long single-colored subnet as shown in Fig. 5(b). The intermediate pin of a net is defined on the grid point whose color is same as the color of the single-colored subnet. In order to increase the length of single-colored subnet while keeping the total length of the net small,



(a) Input (b) Subproblem Definition (c) Fill grid

(d) Mandrel (e) Slimming and Sidewall (f) Final Pattern

Fig. 5. Outline of the proposed approach

the intermediate pin is better to be defined on a nearest grid point of the pin of the double-colored net within the bounding box of the net. Third, for each double-colored subnet, the design of influenced area of the subnet is generated. The influenced area is regarded as obstacle in the following routing pattern generation. Forth, the routing pattern of single-colored nets and single-colored subnets are generated based on the partially pre-colored two-color base grid with obstacle as shown in Fig. 5(b). Fifth, the type of each dummy pattern is decided to create no undesired connections as shown in Fig. 5(c). Finally, each grid point in obstacle and extension pin is recolored in opposite color if the length of routing pattern can be decreased as shown in Fig. 5(c).

B. *The routing graph for single-colored net*

A routing graph to find a feasible route of a single-colored net in partially pre-colored two-color base grid is defined. A single-colored net is either red or blue. The route of a red-net uses red-grids and white-grids, and the route of a blue-net uses blue-grids and white-grids. A white-grid can be used by both the route of a red-net and the route of a blue-net. The red-routing graph for red-nets and the blue-routing graph for blue-nets are defined. The definitions of the red-routing graph and the blue-routing graph are symmetrical. So, the definition of the red-routing graph is focused in the following.

The red-routing graph consists of vertices that correspond to red-grids and edges that correspond to white-grids. The edge that corresponds to a white grid connects vertices that correspond to adjacent red-grids. Each vertex and edge is assigned a positive cost which is dynamically changed during rip-up and reroute procedure. During our proposed algorithm, minimum cost paths are iteratively obtained by A^* algorithm under each cost assignment. The cost of an element is defined as the sum of *base-cost*, *obstacle-cost*, *history-cost*, and *cut-cost*.

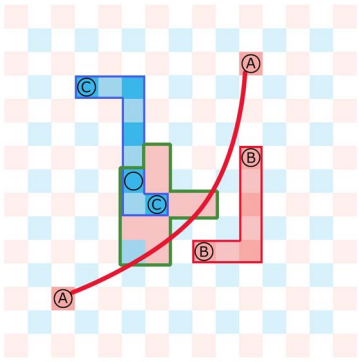


Fig. 6. Example: Connection Request for red net A

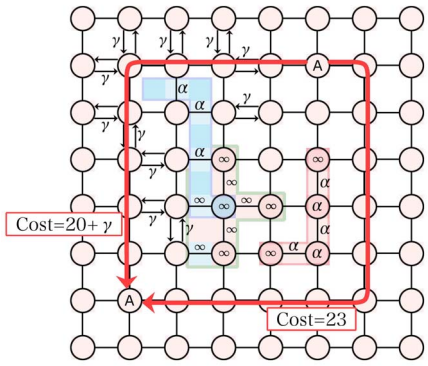
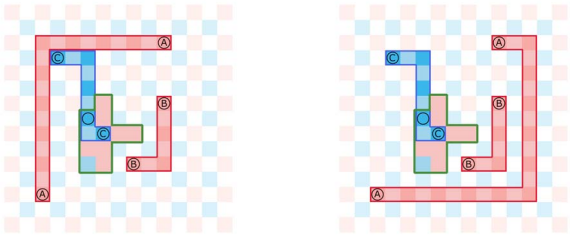


Fig. 7. Routing graph with SADP-aware cost for red net

The base-cost corresponds to the length of a route. The base-cost is 1 in our implementation. A route of a net is not allowed to use a grid to which a pin of other net is assigned or a grid inside influenced area. The obstacle-cost which is ∞ is assigned to an element that corresponds to such a grid. A route of a net is allowed to use a grid which is used by the route of other net, but it is not preferred. The obstacle-cost which is $\alpha > 0$ is assigned to an element that corresponds to a grid which is currently used by the route of other net. The obstacle-cost of the other elements is 0. The history-cost is used in rip-up and reroute procedure. The history-cost of an element is initially set to 0 and increased by β whenever the route at the element is removed. The cut-cost is used to prevent the route of a net from touching the route of a blue-net as much as possible. The cut-cost which is $\gamma > 0$ is assigned to an edge that is connected to a red-grid point which is adjacent to the route of a blue-net, The cut-cost of other elements which include an edge that is connected from a red-grid point which is adjacent to the route of a blue-net is 0.

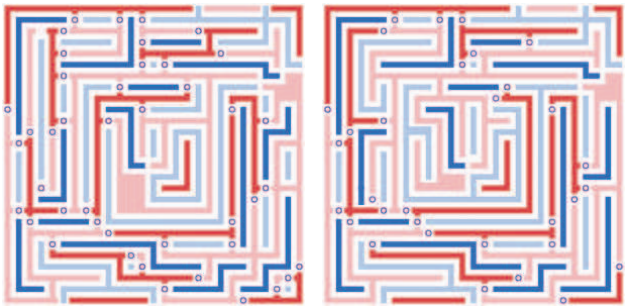
An example of the red-routing graph which corresponds to the situation shown in Fig. 6 is shown in Fig. 7. In this example, the history-cost of each element is assumed to be 0. An edge is a directed edge. In the figure, if the cost of parallel edges in opposite direction is the same, then



(a) Route $\gamma = 0$
(weight = 20 ($\gamma = 0$))
(weight = 30 ($\gamma = 10$))

(b) Route $\gamma = 10$
(weight = 23 ($\gamma = 0$))
(weight = 23 ($\gamma = 10$))

Fig. 8. Route of net A



(a) #Cut-pattern(43) [$\gamma = 0$] (b) #Cut-pattern(32) [$\gamma = 10$]

Fig. 9. Routing Pattern Example

they are shown as an undirected edge for simplicity. The cost of each element is shown in the figure except 1.

C. *Shortest Path Algorithm*

The weight of a route is defined as the sum of costs of grid points and edges in the route. A minimum weight route is selected as the route of a net which is obtained by A^* algorithm in the proposed method. Examples are given in Fig. 8. The routes obtained by setting $\gamma = 0$ and 10 are shown in Fig. 8(a) and (b), respectively. The weight of the route shown in Fig. 8(a) is 20 when $\gamma = 0$, and 30 when $\gamma = 10$. The weight of the route shown in Fig. 8(b) is 23 which is independent of γ.

V. Experimental Result

Experiments are carried out to confirm the effectiveness of our proposed algorithm. The proposed rip-up-and-reroute algorithm is implemented by C++ programming language on Linux PC (CPU:Intel Xeon 2.4 GHz*2). The parameter used in this experiment is as follows. Obstacle-cost, history-cost, and cut-cost are set to $\alpha = 4$, $\beta = 1$ and $\gamma = 10$.

In order to evaluate our routing algorithm, two other routing patterns are obtained by changing the parameters in the cost function of the proposed method. One is generated routing patterns assuming Litho-etch (LE), that is a conventional grid routing, are also generated by changing the parameters of the rip-up-and-reroute algorithm. Both of the color-cost and γ are set to 0 to obtain the reference. The other routing pattern is compliant to SADP process, but obtained without taking the number of cut-pattern into account. The routing pattern is obtained by setting $\gamma = 0$.

The routing pattern of a small problem generated by our implementation is shown in Fig. 9. Routing pattern shown in Fig. 9(a) is obtained by setting $\gamma = 0$ for reference. Routing pattern shown in Fig. 9(b) is an output of our proposed algorithm. In other experiments, testcases used in [7] are used. Note that the layout size of them is the same. Case1 has the lowest routing density, while Case4 has the highest routing density.

The result without taking the number of cut-patterns into account is shown in TABLE I. #Net, #sgl, and #dbl represent the number of nets, the number of single-colored nets, and the number of double-colored nets, respectively. "Total HPWL" is the sum of the half perimeter wire lengths of nets which gives a lower bound of the total wire length. The total wire length, the number of shortest path searches in rip-up-and-reroute, and the CPU time are given in "Total Wire Length", "#Trial", and "CPU", respectively. "LE", "SADP(Gao)", and "SADP($\gamma = 0$)" correspond to results obtained by the method assuming LE, and by the method assuming SADP proposed in [7], respectively. The total wire length of "SADP(Gao)" is from the paper in [7].

As shown in TABLE I, the total wire length and the number of shortest path searches of SADP increase compared to LE. This shows that SADP loses the flexibility of routing compared to LE. The number of shortest path searches and CPU time increase when the routing density increases. CPU time would be reduced much if a faster shortest path algorithm is used. The total wire length of a pattern obtained by the method in [7] is larger than that obtained in [11], Even though the flexibility of routing of each net is larger in [7], routing resource is not fully utilized in global point of view. This shows the validity of the approach proposed in [11] in which the regularity of the base grid is kept as much as possible while eliminating the limitation of connection requirements in [10].

Each result with taking the number of cut-patterns into account with and without Cut-cost is shown in TABLE II. As shown in TABLE II, γ is higher and #Cut-pattern is decreased. However, γ is higher and CPU time is increased. #Cut-pattern and CPU time are relationship of trade-off.

VI. Conclusion

In this paper, SADP-friendly routing algorithm is proposed. A pattern obtained by the proposed algorithm is guaranteed to be manufactured by SADP process if routing on the grid is completed. Experiments show the va-

TABLE I
EXPERIMENTAL RESULT FOR SADP WITHOUT CUT-COST [11]

Testcase (grids)	#Net (#sgl,#dbl)	Total HPWL	Total Wire Length			#Trial		CPU (sec)	
			LE	SADP(Gao)	SADP($\gamma = 0$)	LE	SADP($\gamma = 0$)	LE	SADP($\gamma = 0$)
Case1 (501x501)	300 (263, 37)	3770	3772 (+0.1%)	3820 (+1.3%)	3790 (+0.5%)	302	308 (+2.0%)	0.03	0.05
Case2 (501x501)	600 (528, 72)	7128	7166 (+0.5%)	7330 (+2.8%)	7281 (+2.1%)	656	826 (+25.9%)	0.05	0.30
Case3 (501x501)	800 (717, 83)	9704	9810 (+1.1%)	10130 (+4.4%)	10025 (+3.3%)	1027	1403 (+36.6%)	0.18	0.54
Case4 (501x501)	1000 (891,109)	12171	12289 (+1.0%)	12929 (+6.2%)	12618 (+3.7%)	1292	1942 (+50.3%)	0.25	0.99

TABLE II
EXPERIMENTAL RESULT FOR SADP WITH CUT-COST

Testcase (grids)	#Net (#sgl,#dbl)	Total Wire Length		#Trial		#Cut-pattern		CPU (sec)	
		$\gamma = 0$	$\gamma = 10$	$\gamma = 0$	$\gamma = 10$	$\gamma = 0$	$\gamma = 10$	$\gamma = 0$	$\gamma = 10$
Case1 (501x501)	300 (263, 37)	3790 (+0.5%)	3798 (+0.7%)	308 (+2.0%)	308 (+2.0%)	8	0	0.05	0.08
Case2 (501x501)	600 (528, 72)	7281 (+2.1%)	7325 (+2.8%)	826 (+25.9%)	820 (+25.0%)	50	45	0.30	2.79
Case3 (501x501)	800 (717, 83)	10025 (+3.3%)	10048 (+3.5%)	1403 (+36.6%)	1255 (+22.2%)	96	29	0.54	1.51
Case4 (501x501)	1000 (891,109)	12618 (+3.7%)	12618 (+4.4%)	1942 (+50.3%)	2074 (+60.5%)	159	66	0.99	12.47

lidity of our proposed approach. Farther improvement of quality of routing patterns is in our future works.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (B)25280013.

REFERENCES

[1] C. Bencher, Y. Chen, H. Dai, W. Montgomery, and L. Huli, "22nm half-pitch patterning by CVD spacer self alignment double patterning (SADP)," in *Proc. SPIE, Optical Microlithography XXI*, vol. 6924, 69244E, 2008.

[2] M. C. Smayling, C. Bencher, H. D. Chen, H. Dai, and M. P. Duane, "APF pitch-halving for 22nm logic cells using gridded design rules," in *Proc. SPIE, Design for Manufacturability through Design-Process Integration II*, vol. 6925, 69251E, 2008.

[3] 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," in *Proc. SPIE, Design for Manufacturability through Design-Process Integration*, vol. 6521, 65211K, 2007.

[4] X. Tang and M. Cho, "Optimal layout decomposition for double patterning technology," in *Proc. IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 2011, pp. 9–13.

[5] Y. Kohira, Y. Yokoyama, C. Kodama, A. Takahashi, S. Nojima, and S. Tanaka, "Yield-aware decomposition for LELE double patterning," in *Proc. SPIE, Design-Process-Technology Co-optimization for Manufacturability VIII*, vol. 9053, 90530T, 2014.

[6] M. Mirsaeedi, J. A. Torres, and M. Anis, "Self-aligned double-patterning (SADP) friendly detailed routing," in *Proc. SPIE, Design for Manufacturability through Design-Process Integration V*, vol. 7974, 79740O, 2011.

[7] J.-R. Gao and D. Z. Pan, "Flexible self-aligned double patterning aware detailed routing with prescribed layout planning," in *Proc. ACM International Symposium on Physical Design (ISPD)*, 2012, pp. 25–32.

[8] I.-J. Liu, S.-Y. Fang, and Y.-W. Chang, "Overlay-aware detailed routing for self-aligned double patterning lithography using the cut process," in *Proc. ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2014, pp. 1–6.

[9] F. Nakajima, C. Kodama, H. Ichikawa, K. Nakayama, S. Nojima, T. Kotani, S. Mimotogi, and S. Miyamoto, "Detailed routing with advanced flexibility and in compliance with self-aligned double patterning constraints," in *Proc. SPIE, Design for Manufacturability through Design-Process Integration VII*, vol. 8684, 86840A, 2013.

[10] C. Kodama, H. Ichikawa, K. Nakayama, T. Kotani, S. Nojima, S. Mimotogi, S. Miyamoto, and A. Takahashi, "Self-aligned double and quadruple patterning aware grid routing with hotspots control," in *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2013, pp. 267–272.

[11] T. Ihara, A. Takahashi, and C. Kodama, "Effective two-dimensional pattern generation for self-aligned double patterning," in *Proc. International Symposium on Circuits and Systems (ISCAS)*, 2015, to appear.

[12] Y. Ma, J. Sweis, C. Bencher, H. Dai, Y. Chen, J. P. Cain, Y. Deng, J. Kye, and H. J. Levinson, "Decomposition strategies for self-aligned double patterning," in *Proc. SPIE, Design for Manufacturability through Design-Process Integration IV*, vol. 7641, 76410T, 2010.

[13] Y. Ban, A. Miloslavsky, K. Lucas, S.-H. Choi, C.-H. Park, and D. Z. Pan, "Layout decomposition of self-aligned double patterning for 2D random logic patterning," in *Proc. SPIE, Design for Manufacturability through Design-Process Integration V*, vol. 7974, 79740L, 2011.

[14] Y. Du, Q. Ma, H. Song, J. Shiely, G. Luk-Pat, A. Miloslavsky, and M. D. Wong, "Spacer-is-dielectric-compliant detailed routing for self-aligned double patterning lithography," in *Proc. ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2013, pp. 1–6.