

論文 / 著書情報
Article / Book Information

Title	A Routing Method Using Directed Grid-Graph for Self-Aligned Quadruple Patterning
Authors	Takeshi Ihara, Toshiyuki Hongo, Atsushi Takahashi, Chikaaki Kodama
Citation	IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E100-A, No. 7, pp. 1473-1480
Pub. date	2017, 7
URL	http://search.ieice.org/
Copyright	(c) 2017 Institute of Electronics, Information and Communication Engineers

A Routing Method Using Directed Grid-Graph for Self-Aligned Quadruple Patterning*

Takeshi IHARA[†], Toshiyuki HONGO[†], *Nonmembers*, Atsushi TAKAHASHI^{†a)},
and Chikaaki KODAMA^{††b)}, *Senior Members*

SUMMARY Self-Aligned Quadruple Patterning (SAQP) is an important manufacturing technique for sub 14 nm technology node. Although various routing algorithms for SAQP have been proposed, it is not easy to find a dense SAQP compliant routing pattern efficiently. Even though a grid for SAQP compliant routing pattern was proposed, it is not easy to find a valid routing pattern on the grid. The routing pattern of SAQP on the grid consists of three types of routing. Among them, third type has turn prohibition constraint on the grid. Typical routing algorithms often fail to find a valid routing for third type. In this paper, a simple directed grid-graph for third type is proposed. Valid SAQP compliant two dimensional routing patterns are found effectively by utilizing the proposed directed grid-graph. Experiments show that SAQP compliant routing patterns are found efficiently by our proposed method.

key words: self-aligned quadruple patterning (SAQP), routing algorithm, turn prohibition constraint

1. Introduction

Multiple patterning where features are iteratively formed onto wafer are being investigated as an important manufacturing technique under ArF immersion lithography.

In order to fabricate patterns of 14 nm node, triple patterning lithography (TPL) is intensively paid attention recently. As TPL manufacturing processes, LELELE type in which litho-etch process is repeated three times and LELE-CUT type in which third mask is used to cut the patterns were often discussed in literature. However, all patterns cannot be fabricated by TPL in general. The problem of finding a decomposition of features in LELELE type TPL is NP-hard in general [2]. Even though several decomposition methods including an approximation algorithm for LELELE type TPL [2]–[4] and for LELE-CUT type TPL [5]–[8] were proposed, it is not easy to achieve the target pitch by TPL in practice due to inevitable overlay error.

Self-aligned multiple-patterning is expected to achieve a fine pitch with low process variability. It has no misalignment between pattern-masks since it uses only single litho-etch process for pattern-mask. A fine pitch is achieved by slimming and sidewall spacer process in self-aligned

multiple-patterning. Self-aligned multiple-patterning in which sidewall spacer process is executed once and twice are called *self-aligned double-patterning* (SADP) [9], [10] and *self-aligned quadruple-patterning* (SAQP) [11], respectively, since the pitch achieved by them is half and quarter of the pitch achieved by single litho-etch process, respectively.

SAQP is an important manufacturing technique for sub 14 nm technology node. However the pattern flexibility that can be fabricated on wafer is very limited due to the nature of sidewall spacer process. In SAQP, layout is decomposed into primary, secondary, and tertiary patterns [12], [13]. The width of tertiary patterns is not variable and tertiary patterns have loop structure. Therefore, trimming process is fundamentally required to fabricate a target pattern as tertiary pattern which is typically tree structure.

In order to manufacture a 2D target pattern on wafer by SAQP, SAQP compliant routing pattern generation methods were discussed in [12]–[16]. A layout principle for SAQP was proposed in [12]. An SAQP compliant flexible routing pattern generation algorithm with trim mask definition was proposed in [14]. In [13], [15], a partially colored regular base grid for SAQP is proposed to generate SAQP compliant routing patterns without taking complex constraints into account. The base grid helps to generate complex but regular routing pattern efficiently. Due to the benefit from regularity of base grid, connection requirements are efficiently realized. However, tertiary patterns must satisfy turn prohibition constraints in the base grid.

Turn prohibition constraints imposed on the base grid is illustrated in Fig. 1(a) where a turn written in black is allowed but a turn written in red is prohibited. The path between *S* and *T* shown in Fig. 1(b) is valid, since it uses allowed black turns only, but the path shown in Fig. 1(c) is invalid since it uses a prohibited red turn.

Maze-like routing algorithms often fail to find a valid tertiary pattern by using the base grid when costs are assigned to the base grid in rip-up-and-reroute procedure.

The problem of finding a shortest path under turn prohibition constraints is not easy [17]. Maze-like routing algorithms may not find a valid path even if it exists. In [16], a modified Dijkstra's algorithm to find a valid tertiary pattern was proposed. Even though authors claim that a pattern obtained by the algorithm is a valid, additional operations are required to check the validity and there is no guarantee that a valid pattern is always found. Actually, a decision version of shortest path problem under turn prohibition constraints was

Manuscript received September 14, 2016.

Manuscript revised January 20, 2017.

[†]The authors are with Tokyo Institute of Technology, Tokyo, 152-8550 Japan.

^{††}The author is with Toshiba Corporation, Yokohama-shi, 247-8585 Japan.

*The preliminary version was presented at [1].

a) E-mail: atsushi@ict.e.titech.ac.jp

b) E-mail: chikaaki1.kodama@toshiba.co.jp

DOI: 10.1587/transfun.E100.A.1473

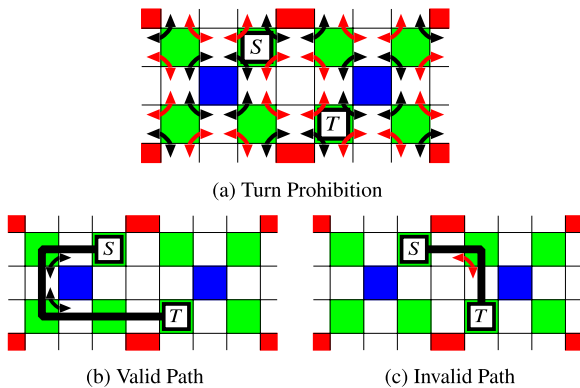


Fig. 1 Turn prohibition constraints on base grid for SAQP (A path can turn on green grid but red turns are prohibited).

proved NP-complete in general [18]. The efforts to design a polynomial time exact shortest path algorithm under turn prohibition constraints would be useless. However, in our tertiary pattern generation, an optimal valid tertiary pattern can be obtained by our method even though turn prohibition constraints are imposed on the base grid.

In this paper, a routing pattern generation method for SAQP is proposed. Our proposed method is based on the base grid for SAQP introduced in [13], [15]. In the proposed method, the primary, secondary, and tertiary routing graphs are defined for primary, secondary, and tertiary patterns of SAQP, respectively. The primary and secondary routing graphs are typical undirected grid-graphs for a routing grid. On the other hand, the tertiary routing graph is defined to match the constraints on tertiary patterns on the base grid. Our proposed method includes a polynomial time exact shortest path algorithm for tertiary patterns of the base grid for SAQP. In [13], [15], no algorithm to obtain a valid tertiary pattern is introduced. Our proposed algorithm obtains an optimal valid tertiary pattern between two pins in terms of a current cost.

The tertiary routing graph is a directed grid-graph in which no prohibited turn is contained in a directed path. In the tertiary routing graph, a valid tertiary pattern between two pins may not correspond to a directed path from a pin to the other pin. But, if so, it corresponds to a directed path from the latter pin to the former pin. Our proposed algorithm finds a shortest path between two pins that satisfies turn prohibition constraint by exploring the graph from both pins.

The tertiary routing graph defined in this paper is different from the graph proposed in the preliminary version of this paper. In the preliminary version [1], the tertiary routing graph contains directed path that forms forbidden tertiary pattern. On the other hand, the tertiary routing graph defined in this paper contains no directed path that forms forbidden tertiary pattern, and its size is roughly a half of the preliminary version.

In the proposed method for SAQP routing, an A^* base shortest path algorithm with rip-up and reroute technique is used to generate routings in primary, secondary, and tertiary

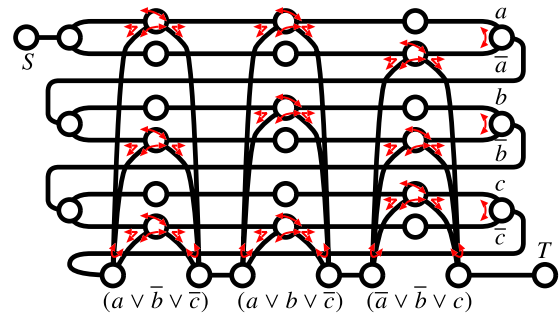


Fig. 2 Polynomial time reduction from 3-SAT (The graph with turn prohibition constraints constructed for 3-SAT instance $(a \vee b \vee \bar{c}) \wedge (a \vee b \vee \bar{c}) \wedge (\bar{a} \vee \bar{b} \vee c)$).

routing graphs. Experiments show that SAQP compliant routing patterns are obtained efficiently.

2. Preliminaries

2.1 Turn Prohibition Constraints

The problem of finding a path that satisfies turn prohibition constraints is formulated as the problem of finding a valid path. Given two vertices and the set of forbidden edge pairs, a path between two vertices is said to be valid if it contains no forbidden pair. Turn prohibition may correspond to traffic rules. Car navigation system must find a route that follows traffic rules, though a route that passes an intersection twice is allowed.

NP-completeness of the problem of valid path existence checking under turn prohibition constraints was proved by showing polynomial time reduction from 3SAT (3-satisfiability) [18]. The details of the proof is omitted, but an example of the problem instance of valid path existence checking defined for a 3-SAT instance is illustrated in Fig. 2.

A graph used in the proof as shown in Fig. 2 contains two types of structures, a structure for variable and a structure for clause. The structure for a variable is a horizontal parallel path structure, and a horizontal path in the structure corresponds to a truth assignment of the variable. The structure for a clause is two vertices that are connected through a vertex in the structure for a variable. A path from source S to sink T that passes structures for all variables and then passes structures for all clauses corresponds to a satisfying truth assignment for the instance. A path that does not follow the intention above is forbade by turn prohibition constraints. Then, there is a valid path from S to T if and only if the instance can be satisfied.

Note that a problem of finding a valid path in the tertiary routing graph defined in Sect. 3.2 can be solved in polynomial time due to the property of graph structure and constraints.

2.2 Self-Aligned Quadruple-Patterning (SAQP) Process

Self-aligned quadruple-patterning (SAQP) is a key manufacturing process for sub 14 nm node technology and beyond

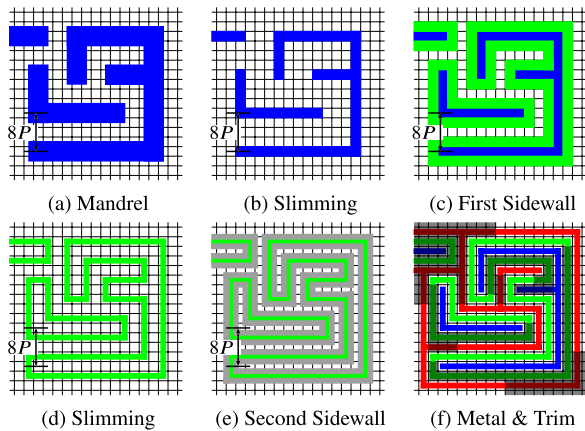


Fig. 3 Self-aligned quadruple-patterning (SAQP) process (half pitch = $1P$, grid pitch = $2P$).

[11]. A typical SAQP process is shown in Fig. 3 where the half pitch of the final pattern is $1P$ and the grid pitch is $2P$. (a) Mandrel of resist pattern is formed by optical lithography where the width of mandrel and the space between mandrels are both $4P$. (b) Mandrel of $2P$ width is obtained by slimming process. (c) The first sidewall of mandrel whose width is $2P$ is formed by depositing masking material and etching back the material. (d) The first sidewall of $1P$ width is obtained by slimming process after removing mandrel. (e) The second sidewall whose width is $1P$ is formed by depositing masking material and etching back the material. (f) The substrate is etched by using the sidewall as a mask, and the sidewall is removed. The etched area is filled by conductive material and the final pattern of $1P$ half pitch is formed by trim.

In SAQP, layout is decomposed into primary, secondary, and tertiary patterns [12], [13]. A stitch which is allowed in LELELE and LELE CUT is not allowed. Primary patterns correspond to mandrel patterns formed on wafer by single exposure. Secondary patterns are formed between every primary patterns after the first sidewall spacer process. Tertiary patterns are formed between primary and secondary patterns after the second sidewall spacer process. The final wafer image of $1P$ half pitch is obtained by trimming the generated patterns. Note that the pitch between primary patterns and the pitch between secondary patterns are both $8P$, while the pitch between tertiary patterns is $4P$. Also, note that multiple patterning techniques are required during trimming process if a fine trimming is needed to be realized.

Among three types of patterns in SAQP, tertiary patterns have several structural restrictions since tertiary patterns correspond to the first sidewall spacer.

First, the width of tertiary patterns is not variable. The width of sidewall spacer is determined based on the quantity of deposit material and it is impossible to partially control the quantity though the deviation of the width is very small.

Second, tertiary patterns have loop structure when sidewall spacer is formed around the patterns of tree structure on the wafer. Therefore, trimming process is fundamentally

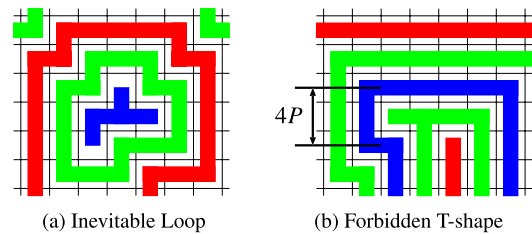


Fig. 4 Tertiary pattern (green) in SAQP.

required to fabricate a target pattern by SAQP which typically consists of tree structures. In the layout pattern shown in Fig. 4(a), tertiary pattern (green) that surrounds primary pattern (blue) forms a loop.

Third, tertiary patterns do not have a T-shape structure. In order to manufacture a T-shape structure of tertiary pattern, the pitch of $4P$ is required in either between primary patterns or between secondary patterns. For example, tertiary pattern (green) shown in Fig. 4(b) contains T-shape and the minimum pitch of primary pattern (blue) is $4P$ which is impossible to manufacture. Note that the pattern cannot be fabricated as secondary pattern as well. Therefore, the connection of a net in tertiary pattern in SAQP is realized as path structure even if the net contains multi pins since no fork is allowed in connection.

2.3 Partially Pre-Colored Three-Color Base Grid

The decomposition of a general layout pattern into primary, secondary, tertiary patterns is not intuitive, and the characterization of general SAQP compliant layout pattern is not easy. In order to derive SAQP compliant layout patterns, a partially pre-colored three-color base grid was proposed in [13], [15] which is shown in Fig. 5. In a layout pattern obtained by using the base grid, the width of primary, secondary and tertiary patterns, and the space between them are identical, which is the half pitch of layout pattern. The pitches of primary patterns, secondary patterns and tertiary patterns are 4 grids, 4 grids, and 2 grids, respectively. Primary patterns and secondary patterns align alternately, and tertiary patterns align between them.

A grid of the base grid whose coordinate is represented by a pair of integers is to be colored to one of blue, red, or green, which correspond to primary, secondary, or tertiary patterns of SAQP, respectively. The connection requirement of a net is satisfied when all pins of the net is connected by grids with the same color. A color is assigned to a grid to realize the connection of a net, but there are several restrictions.

In the base grid, grids whose coordinates are $(2 \bmod 4, 2 \bmod 4)$, $(0 \bmod 4, 0 \bmod 4)$, and $(1 \bmod 2, 1 \bmod 2)$ are pre-colored by blue, by red, and by green, respectively, and are referred by B, R, and G, respectively. The remaining grids are called white grids. A white grid whose coordinate is either $(0 \bmod 4, 2 \bmod 4)$ or $(2 \bmod 4, 0 \bmod 4)$ is to be colored to either blue or red. A white grid whose coordinate is either $(1 \bmod 2, 2 \bmod 4)$ or $(2 \bmod 4, 1 \bmod 2)$ is to be

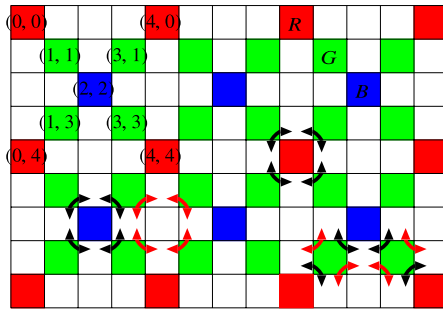


Fig. 5 Base grid for SAQP.

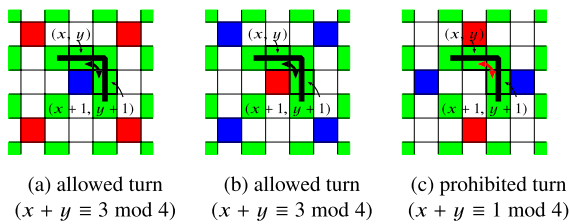


Fig. 6 Turn prohibition constraints of tertiary patterns on base grid.

colored to either blue or green. A white grid whose coordinate is either $(0 \bmod 4, 1 \bmod 2)$ or $(1 \bmod 2, 0 \bmod 4)$ is to be colored to either red or green.

In addition, in order to make tertiary pattern align between primary and secondary patterns, the assignment of green to a white grid is further restricted as follows: At least either of white grids whose coordinates are (x, y) and $(x+1, y+1)$ where $x+y \equiv 1 \bmod 4$ is to be colored to either blue or red; At least either of white grids whose coordinates are (x, y) and $(x+1, y-1)$ where $x-y \equiv 1 \bmod 4$ is to be colored to either blue or red.

These restrictions correspond to turn prohibition constraints of tertiary patterns in the base grid. A turn of a tertiary routing pattern in the base grid whose inside corner is neither R grid nor B grid is prohibited. In Fig. 5, a part of allowed turns and a part of prohibited turns are written in black and in red, respectively. Valid tertiary patterns and an invalid tertiary pattern in the base grid are shown in Fig. 6. An SAQP compliant layout pattern cannot be obtained by assumed typical SAQP process if invalid tertiary patterns are included.

3. SAQP-Friendly Routing

3.1 Overview

In this paper, SAQP compliant routing algorithm that realizes a given connection requirement is proposed.

The proposed routing algorithm utilizes the partially pre-colored three-color base grid which was proposed in [13], [15] and is explained in the previous section. An obtained layout pattern consists of primary, secondary and tertiary patterns of SAQP, and the width of primary, secondary and tertiary patterns, and the space between them are iden-

Step 1: Route tertiary nets in increasing order of the size of bounding-box of a net.

Step 2: Route primary and secondary nets in increasing order of the size of bounding-box of a net.

Step 3: Rip-up a net that shares a grid with others, and reroute the net. Repeat it until no grid is shared by nets. Abort repetition if the number of repetitions of trials reaches to the predetermined number or if no route is found for a net.

Step 4: Rip-up and reroute nets one by one during which the history cost is ignored and no intersection is allowed.

Step 5: Fill vacant grids by dummy patterns.

Fig. 7 SAQP friendly routing algorithm.

tical, which is the half pitch of layout pattern.

A problem instance of our algorithm is assumed to meet the properties of the base grid. A problem instance consists of three types of connection requirements that are to be manufactured as primary, as secondary, or as tertiary patterns which are called primary nets, secondary nets, or tertiary nets, respectively. A pin of a net is placed on a pre-colored grid, and all pins of a net are placed on grids of the same pre-color. Also, since no fork is allowed in tertiary pattern, a multi-pin net assigned to tertiary pattern (green) is divided into two-pin nets in advance so that they form a path-structure. A problem instance that does not meet the requirements above is out of the scope of this paper, and will be handled by enhancements of the proposed algorithm in future works.

The outline of our proposed SAQP friendly routing algorithm is described in Fig. 7.

The proposed algorithm uses the primary, secondary, and tertiary routing graphs. for primary, secondary, and tertiary nets of SAQP, respectively. In Steps 1, 2 and 3, routing patterns are obtained by using A^* base shortest path algorithm with a rip-up and reroute technique. Three routing graphs share grids in the base grid. The tertiary routing graph has a priority since the restrictions imposed on it are severe than the others. So, in Step 1, routing patterns of tertiary nets are generated, then others are generated in Step 2. Routing patterns obtained in Steps 1 and 2 may have conflicts in each routing graph, also may have conflicts among routing graphs. In Step 3, these conflicts are reduced by a rip-up and reroute a net one by one. History cost that is imposed on the grids during negotiation among nets is utilized. A routing pattern obtained may contain redundant detours caused due to the history cost. In Step 4, such redundant detours are eliminated as post-processing of routing. Finally, in Step 5, grids not used for routing are filled by dummy as described in [13], [15], and the primary patterns that consist of mandrels are determined, but the details of dummy fill are omitted in this paper.

3.2 Routing Graphs

A routing pattern is drawn in the base grid by assigning one of blue, red, or green to a grid. In order to generate a routing pattern that realizes connection requirements, three routing

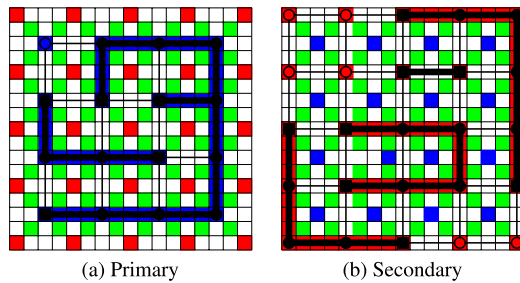


Fig. 8 Routing graphs.

graphs are defined.

Three routing graphs share grids in the base grid. In our proposed algorithm, during exploration of routing, a grid used for the connection of other net is allowed to be used but is not preferred to be used. In order to find a valid routing pattern effectively, during routing pattern exploration, the weight is assigned to each grid of the base grid which is reflected in each routing graph. Each vertex and edge of routing graphs is assigned a positive weight which is dynamically changed during routing pattern exploration.

The primary routing graph, which is an undirected grid-graph, is defined as follows: a vertex corresponds to a B grid; an edge corresponds to three white grids between B grids. The weight of a vertex and the weight of an edge is defined associated with corresponding grids. The secondary routing graph is defined similarly in terms of R grids. Examples of a primary routing graph and a secondary routing graph are shown in Fig. 8(a) and (b), respectively. The routing results that correspond to the layout pattern shown in Fig. 3 are also shown in these figures.

The tertiary routing graph, which is a directed grid-graph, is defined differently to handle restrictions on the tertiary patterns. In the tertiary graph, a vertex corresponds to a G grid. An edge corresponds to a white grid between G grids. The direction of an edge is determined so that a R grid is on the left of it or so that a B grid is on the right of it. Except vertices corresponding to a grid on the boundary of routing area, each vertex has two incoming edges and two outgoing edges. The weight of a vertex and the weight of an edge are defined associated with the corresponding grid. An example of a tertiary routing graph is shown in Fig. 9. The routing result that corresponds to the layout pattern shown in Fig. 3 is also shown in this figure.

In the tertiary graph introduced in the preliminary version [1], each vertex, except vertices corresponding to a grid on the boundary of routing area, has also two incoming edges and two outgoing edges. However, a vertex corresponds to a white grid between G grids, and the number of vertices is roughly twice. An example of a tertiary routing graph in the preliminary version [1] that corresponds to the layout pattern shown in Fig. 3 is also shown in Fig. 10. The graph contains a path that has two edges correspond to a same G grid which corresponds to an invalid tertiary pattern.

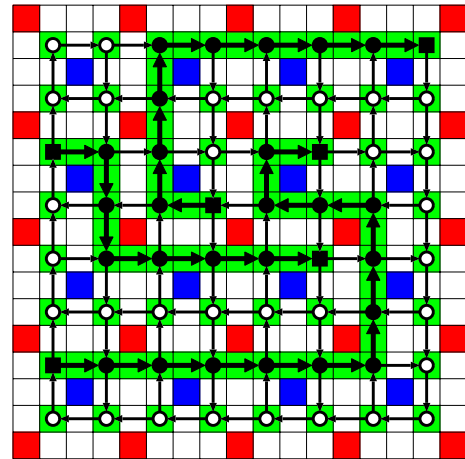


Fig. 9 Tertiary routing graph.

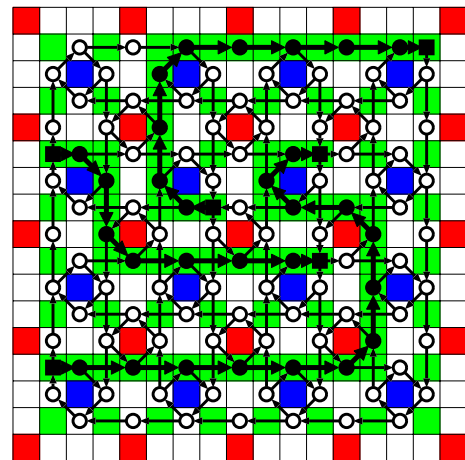


Fig. 10 Tertiary routing graph introduced in [1].

3.3 Valid Path Search in Tertiary Routing Graph

If prohibited turns are simply excluded during path search in the base grid, then there are cases that valid paths cannot be found as shown in Fig. 11(a). Searches for valid paths are blocked by shorter paths that cannot reach the destination in simple maze-like algorithms. On the other hand, a valid path is obtained by simple maze-like algorithms in tertiary routing graph as shown in Fig. 11(b).

In the base grid, there are two types of tertiary routing patterns from a pin. One is a routing pattern which has R grids on the left of it. The other is a routing pattern which has R grids on the right of it. The latter type of paths cannot be found by path exploration from the pin in the tertiary routing graph. However, they are explored by path exploration from the other pin of a net. For example, in Fig. 12(a), a valid path from S to T in the tertiary routing graph contains a detour. There is no valid path without detour from S to T in the tertiary routing graph. In this case, a valid path without detour is found as a valid path from T to S as shown in Fig. 12(b).

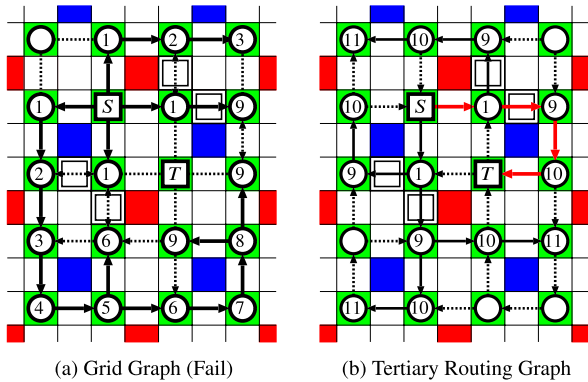


Fig. 11 Path search in base grid. (The cost of a white grid with rectangle = 8, without rectangle = 1. and others = 0 are assumed here. An edge contained in a shortest path from S found during path search is drawn in solid, and other edges are drawn in dotted. The direction of a search expansion is represented by an arrow. In (a), a search that directly violates turn prohibition is excluded in maze search expansion. Search from T also fails by symmetrical property of cost assignment though upper routing area is not shown.)

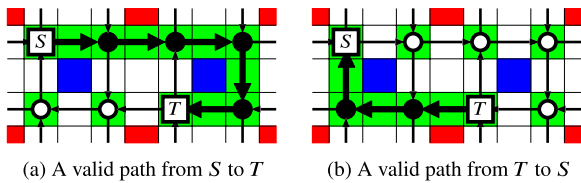


Fig. 12 Path search in tertiary routing graph.

In our implementation, path explorations from both pins are executed individually, and shorter one, which is optimum in terms of current grid weight, is selected. Note that there is a case that a shortest path is not found if two path explorations from both pins are executed simultaneously.

3.4 Grid Weight

The weight of a grid in the base grid is defined as the sum of *base-weight*, *obstacle-weight*, and *history-weight*.

The base-cost corresponds to the length of a route. The base-cost is set to 1 in this paper. A route of a net is not allowed to use a grid to which a pin of other net is assigned. The obstacle-cost which is ∞ is assigned to a grid that corresponds a fixed routing pattern or a forbidden grid.

During rip-up and reroute procedure which corresponds to Steps 1, 2, and 3 in Fig. 7, the obstacle-cost which is $\alpha > 0$ is assigned to a grid that corresponds to a grid which is currently used by the route of other net. The obstacle-cost of the other grids is 0. The history-cost is used in rip-up and reroute procedure. The history-cost of a grid is initially set to 0 and increased by β whenever the route at the grid is removed. In the post-processing of routing which corresponds to Step 4 in Fig. 7, the obstacle-cost which is ∞ is assigned to a grid if it is used by routing pattern of the other nets, while the history-cost is set to 0 for all grids.

Table 1 Testcases on different grid sizes.

Testcase (grids)	#Net (#r, #b, #g)	Total HPWL
Case1 (101x 101)	100 (25, 25, 50)	906
Case2 (241x 241)	500 (174, 160, 166)	6444
Case3 (501x 501)	1500 (451, 449, 600)	19836
Case4 (801x 801)	3000 (911, 889, 1200)	39300
Case5 (1201x 1201)	6000 (1823, 1777, 2400)	78256

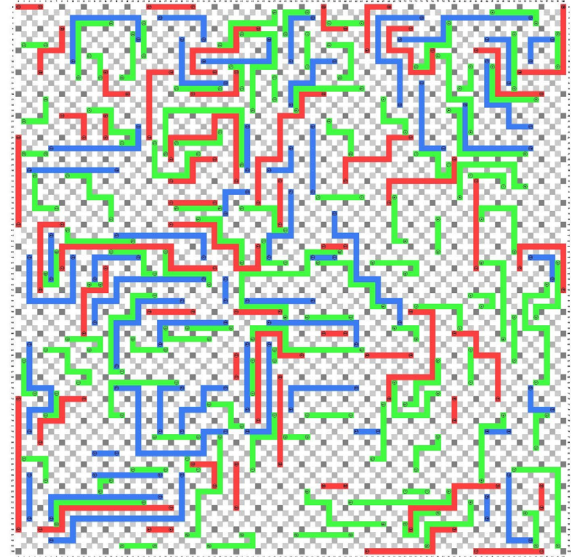


Fig. 13 Routing result (Case1-2).

4. Experiments

Experiments are carried out to confirm the effectiveness of our proposed algorithm. Rip-up-and-reroute algorithm is implemented by C++ programming language on Ubuntu (CPU: Intel Core i7 3.3 GHz). The parameter used in this experiment is as follows. Obstacle-cost and history-cost are set to $\alpha = 4$ and $\beta = 1$, respectively. In testcases used in experiments, two-pin nets are randomly generated so that pins of a net are placed on grids of the same pre-color, and an upper bound of distance between pins are set appropriately.

The route of a net is found by A^* algorithm. A^* algorithm used in experiments is based on “Path-finder” [19]. The weight of a route is defined as the sum of costs of vertices and edges in the route. A minimum weight route is selected as the route of a net which is obtained by the algorithm.

In order to evaluate our routing algorithm, routing patterns assuming litho-etch (LE) are also generated by rip-up-and-reroute algorithm without post-processing in which the conventional routing grid without distinguishing three types of nets is used. An LE-compliant routing pattern is not necessarily to be SAQP-compliant, but it is used to evaluate the impacts of constraints derived by SAQP process.

In Table 1, statistics of testcases on different grid sizes are shown. #Net, #r, #b and #g represent the number of nets, the number of primary nets, the number of secondary nets, and the number of tertiary nets, respectively. “Total HPWL”

Table 2 Experimental result (size).

Testcase	Total Wire Length			#Trial			CPU (sec)		
	LE	SAQP	SAQP+	LE	SAQP	SAQP+	LE	SAQP	SAQP+
Case1	910 (+0.4%)	962 (+6.2%)	962 (+6.2%)	120	108 (−10.0%)	208 (+73.3%)	0.00	0.01	0.04
Case2	6590 (+2.3%)	6892 (+7.0%)	6876 (+6.7%)	688	1104 (+60.5%)	1604 (+133.1%)	0.03	0.34	0.54
Case3	20002 (+0.8%)	20324 (+2.5%)	20304 (+2.4%)	1823	2032 (+11.5%)	3532 (+93.7%)	0.16	1.43	3.27
Case4	39720 (+1.1%)	40512 (+3.1%)	40448 (+2.9%)	3614	4216 (+16.7%)	7216 (+99.7%)	5.03	8.11	20.10
Case5	78962 (+0.9%)	80380 (+2.7%)	80288 (+2.6%)	6932	7940 (+14.5%)	13940 (+101.1%)	3.06	30.58	82.53

Table 3 Experimental result (SAQP, density).

Testcase (grids)	#Net (#r, #b, #g)	Total HPWL	Total Wire Length		#Trial		CPU (sec)	
			SAQP	SAQP+	SAQP	SAQP+	SAQP	SAQP+
Case1-1 (101x 101)	100 (25,25, 50)	906	962 (+6.2%)	962 (+6.2%)	108	208	0.01	0.04
Case1-2 (101x 101)	200 (50,50,100)	2674	2938 (+9.9%)	2930 (+9.6%)	572	772	0.12	0.16
Case1-3 (101x 101)	300 (75,75,150)	4166	4962 (+19.1%)	4910 (+17.9%)	2709	3009	0.67	0.77

is the sum of the half perimeter wire lengths of nets which gives a lower bound of the total wire length. In Table 2, the results for testcases given in Table 1 are shown. 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”, “SAQP”, and “SAQP+” correspond to results obtained by the method assuming LE without post-processing, and by our method assuming SAQP without post-processing, and by our method assuming SAQP, respectively. Note that path explorations from both pins are counted as one shortest path search in “SAQP”, and “SAQP+”. The total wire length and the CPU time of “SAQP” and “SAQP+” are larger than those of “LE,” but it seems to be reasonable. Even though the CPU time is increased by post-processing, the effect of post-processing on reduction of total wire length is confirmed, and better SAQP-compliant routings are obtained.

In Table 3, the SAQP-compliant routing experiments by changing pattern density is shown. Routing result of Case1-2 is shown in Fig. 13. The CPU time and the ratio of the total wire length against total HPWL are increased if the pattern density increases, but it also seems to be a reasonable phenomena.

5. Conclusion

In this paper, an algorithm that generates SAQP compliant pattern efficiently by using a simple directed grid-graph is proposed. In the current implementation of the algorithm, multiple patterning for trim is required. Enhancements of our algorithm for pattern generation for trim is in our future works.

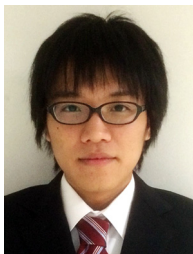
References

- [1] T. Ihara, T. Hongo, A. Takahashi, and C. Kodama, “Grid-based self-aligned quadruple patterning aware two dimensional routing pattern,” *Proc. Design, Automation and Test in Europe (DATE 2016)*, pp.241–244, 2016.
- [2] B. Yu, K. Yuan, B. Zhang, D. Ding, and D.Z. Pan, “Layout decomposition for triple patterning lithography,” *Proc. IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp.1–8, 2011.
- [3] B. Yu, Y.H. Lin, G. Luk-Pat, D. Ding, K. Lucas, and D.Z. Pan, “A high-performance triple patterning layout decomposer with balanced density,” *Proc. IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp.163–169, 2013.
- [4] T. Matsui, Y. Kohira, C. Kodama, and A. Takahashi, “Positive semidefinite relaxation and approximation algorithm for triple patterning lithography,” *Proc. 25th International Symposium on Algorithms and Computation (ISAAC)*, LNCS 8889, pp.365–375, 2014.
- [5] B. Yu, J.R. Gao, and D.Z. Pan, “Triple patterning lithography (TPL) layout decomposition using end-cutting,” *Proc. SPIE, Design for Manufacturability through Design-Process Integration VII*, 8684, 86840G, 2013.
- [6] Y. Kohira, T. Matsui, Y. Yokoyama, C. Kodama, A. Takahashi, S. Nojima, and S. Tanaka, “Fast mask assignment using positive semidefinite relaxation in LELECUT triple patterning lithography,” *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp.665–670, 2015.
- [7] Y. Kohira, C. Kodama, T. Matsui, A. Takahashi, S. Nojima, and S. Tanaka, “Yield-aware mask assignment using positive semidefinite relaxation in LELECUT triple patterning,” *Proc. SPIE, Design-Process-Technology Co-optimization for Manufacturability IX*, 9427, 94270B, pp.1–9, 2015.
- [8] Y. Kohira, C. Kodama, T. Matsui, A. Takahashi, S. Nojima, and S. Tanaka, “Yield-aware mask assignment by positive semidefinite relaxation in triple patterning using cut process,” *J. Micro/Nanolithography, MEMS, and MOEMS (JM3)*, vol.15, no.2-021207, 2016.
- [9] C. Bencher, Y. Chen, H. Dai, W. Montgomery, and L. Huli, “22 nm half-pitch patterning by CVD spacer self alignment double patterning (SADP),” *Proc. SPIE, Optical Microlithography XXI*, 6924, 69244E, 2008.
- [10] M.C. Smayling, C. Bencher, H.D. Chen, H. Dai, and M.P. Duane, “APF pitch-halving for 22 nm logic cells using gridded design rules,” *Proc. SPIE, Design for Manufacturability through Design-Process Integration II*, 6925, 69251E, 2008.
- [11] P. Xu, Y. Chen, Y. Chen, L. Miao, S. Sun, S.W. Kim, A. Berger, D. Mao, C. Bencher, R. Hung, and C. Ngai, “Sidewall spacer quadruple patterning for 15 nm half-pitch,” *Proc. SPIE, Optical Microlithography XXIV*, 7973, 79731Q, 2011.
- [12] K. Nakayama, C. Kodama, T. Kotani, S. Nojima, S. Mimotogi, and S. Miyamoto, “Self-aligned double and quadruple patterning layout principle,” *Proc. SPIE, Design for Manufacturability through Design-Process Integration VI*, 8327, 83270V, 2012.
- [13] 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,” *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp.267–272, 2013.
- [14] F. Nakajima, C. Kodama, H. Ichikawa, K. Nakayama, S. Nojima,

- and T. Kotani, “Self-aligned quadruple patterning-aware routing,” *Proc. SPIE, Design-Process-Technology Co-optimization for Manufacturability VIII*, 9053, 90530C, 2014.
- [15] C. Kodama, H. Ichikawa, K. Nakayama, F. Nakajima, S. Nojima, T. Kotani, T. Ihara, and A. Takahashi, “Self-aligned double and quadruple patterning aware grid routing method,” *IEEE Trans. Computer-Aided Design Integr. Circuits Syst. (TCAD)*, vol.34, no.5, pp.753–765, 2015.
- [16] Y. Ding, C. Chu, and W.K. Mak, “Detailed routing for spacer-is-metal type self-aligned double/quadruple patterning lithography,” *Proc. 52nd Annual Design Automation Conference (DAC)*, 69, 2015.
- [17] E. Gutiérrez and A.L. Medaglia, “Labeling algorithm for the shortest path problem with turn prohibitions with application to large-scale road networks,” *Ann. Oper. Res.*, vol.157, no.1, pp.169–182, 2008.
- [18] T. Hongo and A. Takahashi, “NP-completeness of routing problem with bend constraint,” *IEICE Technical Report, VLD2015-3*, 2015 (in Japanese).
- [19] L. McMurchie and C. Ebeling, “PathFinder: A negotiation-based performance-driven router for FPGAs,” *Proc. FPGA*, pp.111–117, 1995.



Chikaaki Kodama received the B.E., M.E., and D.E. degrees in electronic and information engineering from Tokyo University of Agriculture and Technology, Tokyo, Japan, in 1999, 2001, and 2006, respectively. He was with Fujitsu Ltd., Kawasaki, Japan, from 2001 to 2003, where he worked on development of custom computer-aided design for SPARC64 processor. He is currently with Toshiba Corporation, Yokohama, Japan. His research interests include design for manufacturability, especially mask optimization for optical lithography and VLSI layout design. He is a senior member of IEEE.



Takeshi Ihara received the B.E. and M.E. degrees from the Tokyo Institute of Technology, Tokyo, Japan, in 2014 and 2016, respectively. He is now an engineer at RECRUIT from 2016.



Toshiyuki Hongo received the B.E. and M.E. degrees from the Tokyo Institute of Technology, Tokyo, Japan, in 2014 and 2016, respectively. He is now an engineer at RICOH from 2016.



Atsushi Takahashi received the B.E., M.E., and D.E. degrees in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1989, 1991, and 1996, respectively. He was at the Tokyo Institute of Technology as a Research Associate from 1991 to 1997, and as an Associate Professor from 1997 to 2009 and from 2012 to 2015. From 2009 to 2012, he was at Osaka University, Suita, Japan, as an Associate Professor. He visited University of California, Los Angeles, U.S.A., as a Visiting Scholar from 2002 to 2003. He is currently a Professor with Department of Information and Communications Engineering, School of Engineering, Tokyo Institute of Technology. His current research interests include VLSI layout design and combinational algorithms. He is a senior member of IEEE and IPSJ, and a member of ACM.