Grid-based Self-Aligned Quadruple Patterning Aware Two Dimensional Routing Pattern

Takeshi Ihara, Toshiyuki Hongo, Atsushi Takahashi
Tokyo Institute of Technology, Tokyo 152–8550
Email: {ihara,hongo,atsushi}@eda.ce.titech.ac.jp

Chikaaki Kodama
Toshiba Corporation, Kanagawa 247–8585
Email: chikaaki1.kodama@toshiba.co.jp

***Abstract*—A routing grid for Self-Aligned Quadruple Patterning (SAQP) helps to find a valid routing of SAQP, but it is not easy to find it. The routing 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, SAQP compliant two dimensional routings are found on the grid by finding an optimal valid tertiary routing effectively. Experiments show that SAQP compliant routings are found efficiently.**

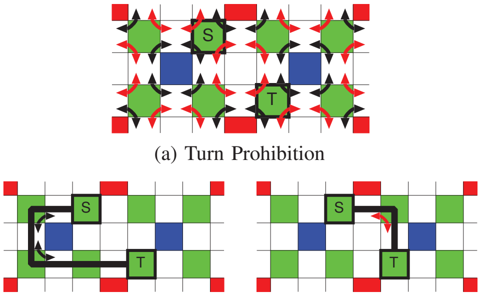
I. Introduction

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. However, all pattern cannot be fabricated by TPL in general. The problem of finding a decomposition of features in LELELE type TPL is NP-hard in general [1]. Even though several decomposition methods including an approximation algorithm for LELELE type TPL [1]–[3] and for LELECUT type TPL [4]–[6] 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) [7], [8] and *self-aligned quadruple-patterning* (SAQP) [9], 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 order to manufacture a 2D target pattern on wafer by SAQP, SAQP compliant routing pattern generation methods were discussed in [10]–[14]. In [11], [13], 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. However, patterns must satisfy turn prohibition constraints.



(a) Turn Prohibition

(b) Valid Path

(c) Invalid Path

Fig. 1. Turn Prohibition Constraints on Base Grid for SAQP (A path can turn on green grid but red turns are prohibited)

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, but the path shown in Fig. 1(c) is invalid. Maze-like routing algorithms often fail to find a valid 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 [15]. In [14], a modified Dijkstra's algorithm to find a valid pattern was proposed. However, there is no guarantee that a valid pattern is always found. Actually, a decision version of shortest path problem under turn prohibition constraints was proved NP-complete in general [16]. However, in our pattern generation, an optimal valid pattern can be obtained by our algorithm even though turn prohibition constraints are imposed on the base grid.

In this paper, a polynomial time exact shortest path algorithm for tertiary patterns of the base grid for SAQP is introduced. An optimal valid tertiary pattern between two pins in terms of the current cost is obtained by the algorithm. In the proposed routing pattern generation method for SAQP, an A^* base shortest path algorithm with rip-up and reroute technique is used to generate routings in primary, secondary, and tertiary routing graphs. Experiments show that SAQP compliant routing patterns are obtained efficiently.

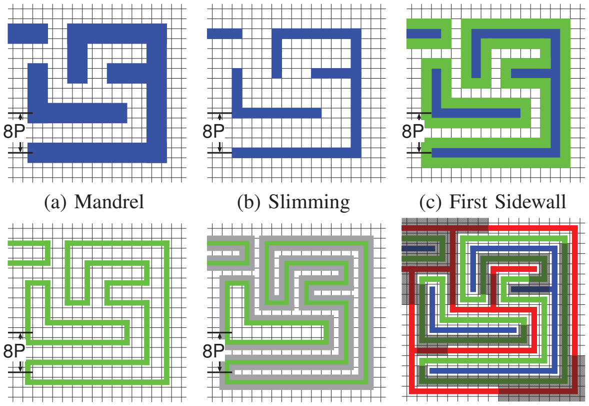
II. Preliminaries

A. Self-aligned quadruple-patterning (SAQP) process

SAQP is a key manufacturing process for sub 14 nm node technology and beyond [9]. A typical SAQP process is shown in Fig. 2 where the half pitch of the final pattern is $1P$ and

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

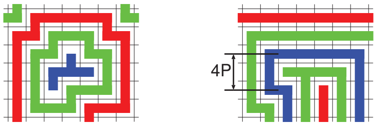
978-3-9815370-6-2/DATE16/©2016 EDAA



(a) Mandrel (b) Slimming (c) First Sidewall

(d) Slimming (e) Second Sidewall (f) Metal & Trim

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



(a) Inevitable Loop (b) Forbidden T-shape

Fig. 3. Tertiary Pattern (Green) in SAQP

the grid pitch is $2P$. In SAQP, layout is decomposed into primary, secondary, and tertiary patterns [10], [11]. 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.

Among three types of patterns in SAQP, tertiary patterns have several structural restrictions. First, the width of tertiary patterns is not variable. Second, tertiary patterns have loop structure. In the layout pattern shown in Fig. 3(a), tertiary pattern (green) that surrounds primary pattern (blue) forms a loop. Third, tertiary patterns do not have a T-shape structure. For example, tertiary pattern (green) shown in Fig. 3(b) contains T-shape and the minimum pitch of primary pattern (blue) is $4P$ which is impossible to manufacture.

B. Partially pre-colored three-color base grid

A partially pre-colored three-color base grid was proposed in [11], [13] which is shown in Fig. 4. A grid of the base grid is to be colored to either blue, red, or green, which correspond to primary, secondary, or tertiary patterns of SAQP, respectively. A color is assigned to a grid to realize the connection of a net, but there are several restrictions.

Several grids are pre-colored in the base grid. Grids pre-colored by blue, by red, and by green 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 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.

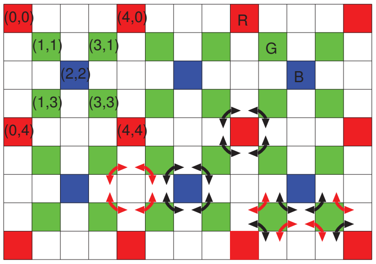
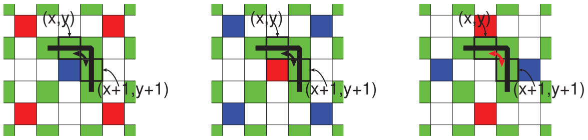


Fig. 4. Base Grid for SAQP



(a) allowed turn $(x + y \equiv 3 \bmod 4)$ (b) allowed turn $(x + y \equiv 3 \bmod 4)$ (c) prohibited turn $(x + y \equiv 1 \bmod 4)$

Fig. 5. Turn Prohibition Constraints of Tertiary Patterns on Base Grid

In addition, 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. Valid tertiary patterns and an invalid tertiary pattern in the base grid are shown in Fig. 5.

III. SAQP-friendly Routing

A problem instance of our algorithm is assumed to meet the properties of the base grid. The outline of our proposed SAQP friendly routing algorithm is described in Fig. 6.

In order to generate a routing pattern that realizes connection requirements, three routing graphs are defined. 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.

The primary routing graph, which is an undirected 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

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:Fill vacant grids by dummy patterns.

Fig. 6. SAQP Friendly Routing Algorithm

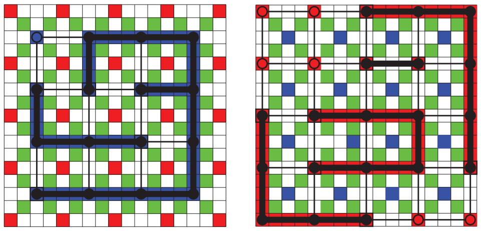

(a) Primary (b) Secondary

Fig. 7. Routing Graphs

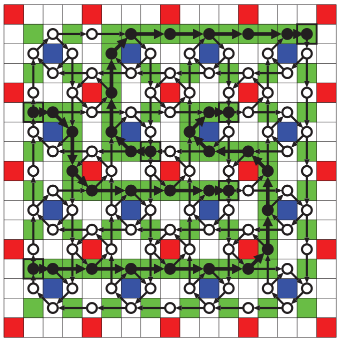
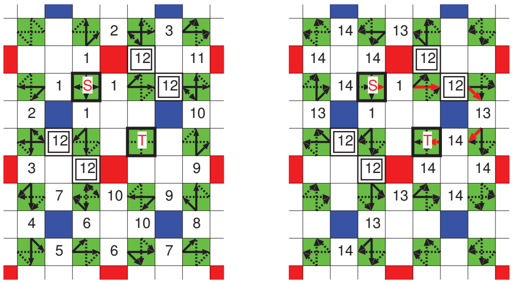


Fig. 8. Tertiary Routing Graph

similarly in terms of R grids. Examples of a primary routing graph and a secondary routing graph are shown in Fig. 7(a) and (b), respectively.

The tertiary routing graph is a directed graph. In the tertiary graph, a vertex corresponds either to a white grid between G grids or to G grid where a pin is assigned. For each G grid to which a pin is not assigned, edges are defined between vertices corresponding to white grids which are adjacent to the G grid except pair corresponding to turn prohibition. For each G grid to which a pin is assigned, edges are defined between the vertex corresponding to the G grid and vertices corresponding to adjacent white grids. The direction of an edge is determined so that a R grid is on the left of it. The weight of a vertex and the weight of an edge are defined associated with the corresponding white grid and G grid, respectively. An example of a tertiary routing graph is shown in Fig. 8.

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. 9(a). While a valid path is obtained by simple maze-like algorithms in tertiary routing graph as shown in Fig. 9(b). In the following, we show that a routing pattern that satisfies the turn prohibition on the base grid is obtained by finding a shortest path of the tertiary routing graph.



(a) Extended Grid Graph (Fail) (b) Tertiary Routing Graph

Fig. 9. Path Search in Base Grid. (The cost of a white grid with rectangle = 11, without rectangle = 1. and others= 0 are assumed here. An edge in search tree is solid, not in search tree is dotted. In (a), a search that directly violates turn prohibition and a search that causes U-turn are excluded. Search from T also fails by symmetrical property though upper routing area is not shown.)

If each edge of a directed path corresponds to a distinct G grid of the base grid, then the corresponding pattern satisfies the constraints imposed on a tertiary pattern. Here, we show that each edge of a shortest path corresponds to a distinct G grid of the base grid. Assume contrary that there is a shortest path P that contains sub-path from a to d where two distinct edges (a, b) and (c, d) correspond to the same G grid. Then, there is a directed edge (a, d) in the tertiary routing graph. Also, $w(a, d) < w(a, b) + w(c, d)$ where $w(u, v)$ is the weight of edge (u, v) since the weight of a grid is positive and the weight of an edge defined associated with the corresponding G grid. Moreover, the weight of the sub-path from a to d of P is positive. Then, the weight of directed path P' which is obtained from P by replacing the sub-path from a to d of P with edge (a, d) is smaller than that of P, and contradicting the assumption that P is shortest. Therefore, each edge of a shortest path corresponds to a distinct G grid of the base grid.

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.

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. The obstacle-cost which is ∞ is assigned to a grid that corresponds a fixed routing pattern or a forbidden grid. 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.

IV. Experiments

The proposed algorithm is implemented by C++ on Ubuntu (CPU:Intel Core i7 3.3 GHz). Obstacle-cost and history-cost are set to $\alpha = 4$ and $\beta = 1$, respectively. A^* algorithm used is based on "Path-finder" [17]. The weight of a route is defined as the sum of costs of vertices and edges in the route. Routing patterns assuming litho-etch (LE) are also generated in which the conventional routing grid is used.

The result is shown in Table I and Table II. #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" is the sum of the half perimeter wire lengths of nets. 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" correspond to results obtained by the method assuming LE, and by our method assuming SAQP, respectively. Routing result of Case1-2 is shown in Fig. 10.

TABLE I
EXPERIMENTAL RESULT (SIZE)

Testcase (grids)	#Net (#r, #b, #g)	Total HPWL	Total Wire Length		#Trial		CPU (sec)	
			LE	SAQP	LE	SAQP	LE	SAQP
Case1 (101x 101)	100 (25, 25, 50)	906	910 (+0.4%)	958 (+5.7%)	120	112 (−6.7%)	0.00	0.00
Case2 (241x 241)	500 (174, 160, 166)	6444	6590 (+2.3%)	6908 (+7.2%)	688	1097 (+59.4%)	0.03	0.31
Case3 (501x 501)	1500 (451, 449, 600)	19836	20002 (+0.8%)	20340 (+2.5%)	1823	1949 (+6.9%)	0.16	3.70
Case4 (801x 801)	3000 (911, 889,1200)	39300	39720 (+1.1%)	40516 (+3.1%)	3614	4153 (+14.9%)	5.03	12.40
Case5 (1201x 1201)	6000 (1823,1777,2400)	78256	78962 (+0.9%)	80420 (+2.8%)	6932	7738 (+11.6%)	3.06	13.48

TABLE II
EXPERIMENTAL RESULT (SAQP, DENSITY)

Testcase (grids)	#Net (#r,#b, #g)	Total HPWL	Total Wire Length	#Trial	CPU (sec)
Case1-1 (101x 101)	100 (25,25, 50)	906	958 (+5.7%)	112	0.00
Case1-2 (101x 101)	200 (50,50,100)	2674	2930 (+10.2%)	508	0.05
Case1-3 (101x 101)	300 (75,75,150)	4166	4986 (+19.7%)	2720	0.18

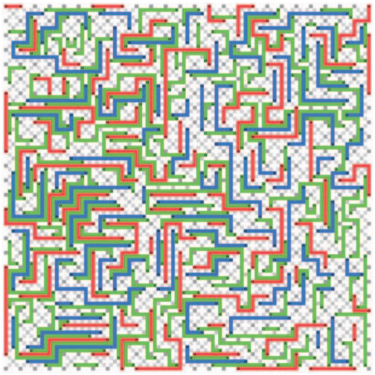


Fig. 10. Routing Result (Case1-2)

The litho-etch and SAQP-compliant routing result by changing grid size is shown in Table I. The SAQP-compliant routing result by changing pattern density is shown in Table II. The total wire length and the computation time are increased compared to LE. However it seems to be affordable in spite of the tight constraints in SAQP.

V. CONCLUSION

In this paper, an algorithm that generates SAQP compliant pattern efficiently 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] B. Yu, K. Yuan, B. Zhang, D. Ding, and D. Z. Pan, “Layout decomposition for triple patterning lithography,” in *Proc. IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 2011, pp. 1–8.

[2] 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,” in *Proc. IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 2013, pp. 163–169.

[3] T. Matsui, Y. Kohira, C. Kodama, and A. Takahashi, “Positive semidefinite relaxation and approximation algorithm for triple patterning lithography,” in *Proc. the 25th International Symposium on Algorithms and Computation (ISAAC), LNCS 8889*, 2014, pp. 365–375.

[4] B. Yu, J.-R. Gao, and D. Z. Pan, “Triple patterning lithography (TPL) layout decomposition using end-cutting,” in *Proc. SPIE, Design for Manufacturability through Design-Process Integration VII*, vol. 8684, 86840G, 2013.

[5] 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,” in *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2015, pp. 665–670.

[6] Y. Kohira, C. Kodama, T. Matsui, A. Takahashi, S. Nojima, and S. Tanaka, “Yield-aware mask assignment using positive semidenite relaxation in LELECUT triple patterning,” in *Proc. SPIE, Design-Process-Technology Co-optimization for Manufacturability IX*, vol. 94270B, 2015, pp. 1–9.

[7] 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.

[8] 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.

[9] 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 15nm half-pitch,” in *Proc. SPIE, Optical Microlithography XXIV*, vol. 7973, 79731Q, 2011.

[10] K. Nakayama, C. Kodama, T. Kotani, S. Nojima, S. Mimotogi, and S. Miyamoto, “Self-aligned double and quadruple patterning layout principle,” in *Proc. SPIE, Design for Manufacturability through Design-Process Integration VI*, vol. 8327, 83270V, 2012.

[11] 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.

[12] F. Nakajima, C. Kodama, H. Ichikawa, K. Nakayama, S. Nojima, and T. Kotani, “Self-aligned quadruple patterning-aware routing,” in *Proc. SPIE, Design-Process-Technology Co-optimization for Manufacturability VIII*, vol. 9053, 90530C, 2014.

[13] 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 Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 34, no. 5, pp. 753–765, 2015.

[14] Y. Ding, C. Chu, and W.-K. Mak, “Detailed routing for spacer-is-metal type self-aligned double/quadruple patterning lithography,” in *Proc. the 52nd Annual Design Automation Conference (DAC)*, no. 69, 2015.

[15] E. Gutiérrez and A. L. Medaglia, “Labeling algorithm for the shortest path problem with turn prohibitions with application to large-scale road networks,” *Annals of Operations Research*, vol. 157, no. 1, pp. 169–182, 2008.

[16] T. Hongo and A. Takahashi, “NP-completeness of routing problem with bend constraint,” in *IEICE Technical Report*, vol. 115, no. 21, VLD2015-3, 2015, pp. 13–18, in Japanese.

[17] L. McMurchie and C. Ebeling, “PathFinder: A negotiation-based performance-driven router for FPGAs,” in *Proc. FPGA*, 1995, pp. 111–117.