Double-Patterning Friendly Grid-Based Detailed Routing with Online Conflict Resolution

Islam S. Abed
Design-to-Silicon Division,
Mentor Graphics Egypt
Cairo, Egypt
islam_shaker@mentor.com

Amr G. Wassal
Computer Engineering Department,
Cairo University
Cairo, Egypt
a.wassal@ieee.org

***Abstract*—Double patterning lithography (DPL) is seen as one of the most promising solutions for new technology nodes such as 32nm and 22nm. However, DPL faces the challenges of handling layout decomposition and overlay errors. Currently, most DPL solutions use post-layout decomposition which requires multiple iterations and designer intervention to achieve a decomposable layout as designs scale larger. Recent research is starting to consider DPL constraints during the layout design phase especially during the detailed routing phase. In this work, we propose DPL-aware grid-based detailed routing algorithm supported with online conflict resolution. The conflict resolution algorithm uses a graph structure to represent geometrical relations between routed polygons and helps in conflict detection and color assignment. Experimental results indicate that this enhanced algorithm reduces the number of conflicts by 60% on average.**

Keywords- Double Patterning Lithography, grid-based, detailed routing, conflict resolution.

I. INTRODUCTION

As the semiconductor industry moves towards deeper nodes such as 32nm and 22nm, lithography technologies are challenged to keep pace. The current 193nm wavelength immersion lithography reached its printing limit of 40 nm half pitch. At the same time, Extreme Ultra Violet (EUV) technology is facing problems getting to production because of the availability of EUV source and inspection equipments. On the other hand, Double Patterning Lithography (DPL) [1] is gaining more attention as a promising manufacturing solution for 32nm and 22nm.

In DPL, each single layout layer can be decomposed into two masks, and manufactured in two exposure and etching steps. This procedure helps decreasing the feature density in each exposure as well as improving resolution and depth of focus (DOF). This extends the use of current immersion lithography equipment to the production of smaller technology nodes. Fig. 1, illustrates the basic idea of layout decomposition.

The main challenges for DPL technologies are efficient layout decomposition of large designs and handling overlay errors. Layout decomposition is the process of assigning different colors to layer features within a distance smaller than the minimum spacing distance for the process technology. A stitch is created while splitting a polygon in two parts with different colors to resolve conflicts in the design as shown in Fig. 2. Due to complex geometric relationships between the patterns in today's typical dense and large designs, successful decomposition cannot usually be achieved without manual intervention possibly for several iterations. To make matters worse, some conflicts are irresolvable and cannot be resolved by splitting conflicting polygons even after manual and time-consuming design modifications. This reduces the efficiency of the DPL decomposition process and increases the design cycle. Overlay errors are due to misalignments between the two masks or the two lithographic steps for the same layer. Overlay errors are usually evident after in stitches or between polygons created by the two lithographic steps. These overlay errors affect the quality of the design, yield and cost. Therefore, a lot of research is currently focused on finding high quality decomposition solutions with minimum irresolvable conflicts and stitches to save design time and manufacturing cost.

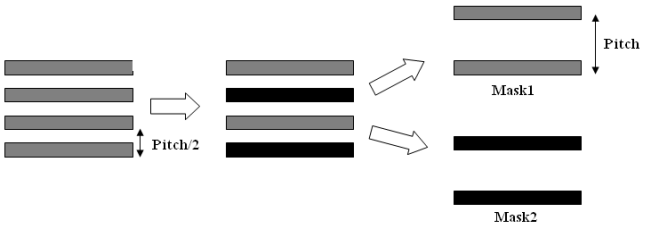

Figure 1. DPL decomposes each layout layer into two masks

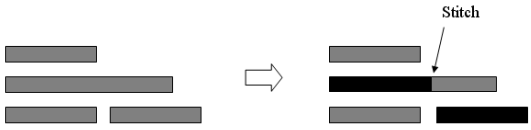

Figure 2. Resolution of DPL conflicts by splitting polygons and adding stitches between the resulting two parts

Most researchers worked on applying decomposition algorithms after the physical design phase or layout is complete [2-7]. Just recently, researchers started to consider DPL decomposition during physical design phase [8-10], especially during detailed routing.

Cho et al [8] were of the first to consider a DPL friendly grid-based detailed routing algorithm. The main idea of their algorithm is to assign a color state for each grid in the routing space and use these grid states to detect conflicts and stitches. The algorithm controls the router decisions by adding conflict and stitching penalties to the routing cost function. After a net is routed, its paths are colored and the surrounding grids are shadowed. In the shadowing process, the coloring states of

978-3-9810801-8-6/DATE12/©2012 EDAA

surrounding grids of a routed path are changed based on the DPL spacing constraints. Although this algorithm achieves remarkable results as reported in [8], it has two main weaknesses. First, it is a greedy algorithm because it searches for the best solution for the current net, regardless of the effect on other nets in the queue. Second, it uses preemptive coloring for the routed paths where a path is immediately colored after it has been completed. This algorithm is revisited in section II. Gao et al [9] tried to enhance the algorithm introduced in [8]. They try to address the problem of resolving improper coloring assignments by delaying the coloring and shadowing process until they have enough information about the surrounding paths. This leads to better routing results and positively enhances the number of stitches. This work also introduced a technique to detect conflicts in candidate paths in the route searching phase. The techniques proposed in [9] were not able to resolve conflicts during routing, and post-routing DPL techniques are needed for conflict resolution. Yen-Hung et al in [10] introduced the first grid-less DPL friendly detailed routing algorithm. The algorithm is based on maintaining a conflict graph for the routed nets, and the evaluation of the cost for each candidate routing path during each routing propagation step. The algorithm has the ability to resolve generated conflicts during routing by stitching or flipping the colors of routed paths. However, it needs to examine the conflict graph with each routing propagation step adding overhead to the routing process and bloating the run time.

In this paper, we use a conflict graph structure along with the DPL friendly grid-based detailed routing algorithm described in [8], and use an online conflict resolution algorithm exploiting the conflict graph to decrease the number of conflicts. In this paper, section II briefly introduces the basic idea of the DPL-friendly detailed routing algorithm in [8] and reviews its limitations. In section III, we introduce an enhanced conflict graph data structure and discuss its characteristics. Section IV presents the online conflict resolution algorithm. Experimental results are described in section V and conclusions are finally given in section VI.

II. DPL-FRIENDLY DETAILED ROUTING ALGORITHMS

In this section, we will review the DPL-friendly detailed grid based routing algorithms [8] for problem formulation. The work introduced one of the first grid-based DPL friendly detailed routing algorithms where each grid on the routing graph gets a state to define its expected color, as shown in Table I.

TABLE I. GRID STATE DEFINITION

Grid state	Description	Grid Color
BG	Bi-colorable	Black or Gray
$B\overline{G}$	Black-colorable	Black
$\overline{B}G$	Gray-colorable	Gray
$\overline{BG}$	Uncolorable	No color

The algorithm starts by assigning the bi-colorable state to all routing grids. Then, a layout decomposition algorithm is applied to assign colors to already existing blocks, such as, clock signals and terminals. The next step is to assign states to grids surrounding the routed blocks within a distance less than the minimum spacing distance. The detailed routing algorithm starts to search for suitable path for each net and considers DPL constraints while calculating the cost of each grid. After finding the paths for each net, the algorithm starts to assign colors to each grid in the path by using grid coloring states. The $B\overline{G}$ states will be assigned to black, while the $\overline{B}G$ states will get the gray color, BG states will be assigned according to the nearest colored grid. The final step is to shadow around colored paths. Fig. 3, illustrates an example of how the algorithm works.

The router adds a stitching penalty to adjacent grids in the same path with different states. Moreover, it adds a larger penalty in case of conflicts. Experimental results for this algorithm are very promising. However, the algorithm is a greedy one that assigns colors to paths according to their routing order in a preemptive manner.

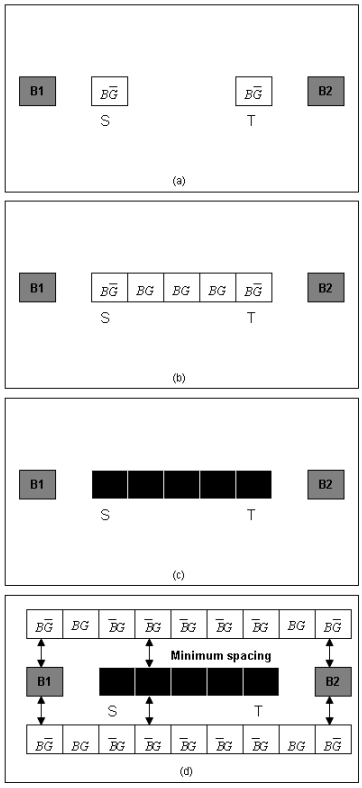

Fig. 3. An example illustrating DPT friendly detailed routing algorithm

III. THE PROPOSED ENHANCED CONFLICT GRAPH DATA STRUCTURE

Conflict graphs are used in DPL decomposition to maintain geometrical relationships between wire segments in a layout [2, 5-7, 11]. Once a conflict is detected using a conflict graph, different techniques can be applied to resolve such conflicts. A conflict graph is defined as CG = *(V, E)*, where *V* represents the routed polygons in a layout, and *E* represents the geometrical relationship between two vertices. The conflict graph is composed of a set of conflict sub-graphs (CSG), where CSG $\subset$ CG.

This work proposes an Enhanced Conflict Graph (ECG), such that ECG = (V^R, E), where V^R denotes the graph vertex set. The vertex set V^R is presented as $V^R(c,n)$, where c is the possible color to be assigned for a polygon, and n represents the number of polygons will be affected if the color c changed for this polygon. The number of affected polygons n for a vertex V^R_i is equal to the summation of the affected polygons for each vertex V^R_j, where V^R_i and V^R_j belongs to the same sub-graph, and have common edge.

$$V^R(n)_i = \sum_{j=1}^{N} V^R(n)_j \quad (1)$$

The ECG is created during routing and a vertex V^R will be added to the graph with each generated path in the routing step. An edge will connect two vertices if the spacing distance between the two paths is less than the double patterning minimum spacing distance. Each vertex in the graph will represent a routed polygon, and the associated information {c, n} facilitates decision making for double patterning conflict resolution. Fig. 4, Shows an example for ECG creation during routing, and the assign of c, and n to each vertex that help in judging the best option for conflict resolution. In some cases, the conflict can be resolved by flipping the color state of one vertex connected to the conflicting one, as explained in the next section. Other cases can be resolved by splitting a vertex into two vertices with different colors and in separate sub-graphs.

IV. Online Conflict Resolution Algorithm

The online conflict resolution algorithm (OCRA) is implemented to tackle the two main limitations in the DPL friendly detailed routing algorithm defined in [8] which is based on routing order and the immediate color assignment for routed polygons. Fig. 5, gives an example of the effect of routing order on the creation of conflicts while using the DPL-friendly detailed routing algorithm. Polygon (A) is a path for net 1. Polygons (B), (C), (D) and (E) have been generated while searching for paths belongs to nets 2, 3 and 4. As defined in the DPL-friendly routing algorithm, the routed path will be colored immediately to Gray with a shadow around it. The immediate coloring and shadowing affect polygon (B) and will be assigned to Black. The same applies for polygons (C), (D) and (E). While searching for possible paths for net 5, the current coloring scheme prohibits the usage of mandatory grids for routing this net which causes a conflict for polygon (F). The online conflict resolution algorithm tries to resolve coloring conflicts during detailed routing assisted by the enhanced ECG.

The algorithm examines each conflict vertex and checks if it is possible to flip the color state of one of the connected vertices. The flipping decision is taken according to the number of affected polygons n assigned to each vertex V^R connected to the conflicting vertex. The algorithm searches for the vertex with the smallest n value to flip and updates the routing space accordingly. If n in all connected vertices exceeds a certain threshold, the algorithm will report a conflict to the designer.

The OCRA algorithm is shown in Fig. 6. The difference between this algorithm and the one defined in [8] is evident in lines 5 to 9 and line 11.

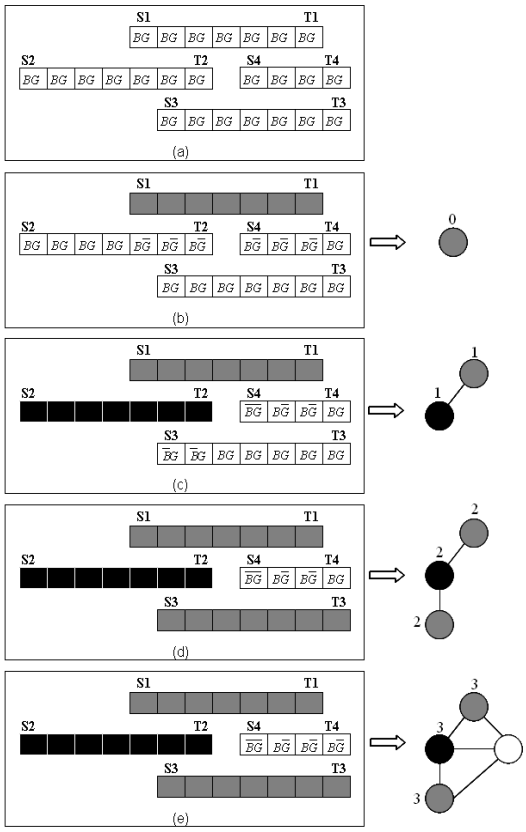

Figure 4. An example of the creation of an enhanced conflict graph (ECG)

The algorithm checks if the generated path will cause conflicts in line 5. In line 6, the connected vertices are examined for flipping. If possible, the flipping algorithm is applied in line 7; otherwise, a conflict node is reported to the designer in line 9. Line 11 adds a vertex to the ECG with each routing step. Reported conflicts can either be resolved using post-layout processing as proposed in [2, 4-7, 11], or could be an irresolvable conflict that requires change in the design.

Fig. 7, shows the ECG corresponds to the layout defined in figure 5 and the resulting graph after conflict resolution. The number associated with each vertex identifies the number n of affected vertices. As shown in the figure, the algorithm will try to resolve the conflict vertex (F) by examining the connected vertices (B) and (D), and choosing the vertex with the smallest n value to flip. The results of the online conflict resolution algorithm are shown in Fig. 8.

Figure 5. An example of the effect of routing order on the results of DPL friendly detailed routing algorithm

The OCRA with DPL Friendly Detailed Routing algorithm is described below:

DPL Friendly Detailed Routing with Online Conflict Resolution Algorithm:

1. Apply layout decomposition for routed blockages, and apply color shadowing
2. For each net *m*
3. Find a path for *m*, and add conflict and stitching penalties to the cost function
4. Assign a grid state to each grid to identify potential color
5. If path *p* has grid in conflict state then
6. If a simple vertex with potential for flipping is found then
7. Flip vertices
8. Else
9. Report conflict
10. Shadow surrounding grids for all affected nodes
11. Add path to conflict graph
12. End For

Figure 6. Double Paternning Friendly Detailed Routing With Online Conflict Resolution Algorithm (OCRA)

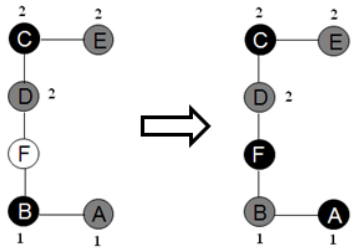

Figure 7. An illustration of a conflict graph representing sample layout data

Figure 8. Results of the online conflict resolution algorithm

V. EXPERMINTAL RESULTS

The presented algorithm has been implemented in C++ and tested on a workstation with a dual-core 2.5GHz processor running a Linux operating system, and populated with 4GB of ram. Four test cases from the examples supplied with the *Alliance Cad System* design suite are used [12]. The critical dimension in these examples is 130nm. Routing is done using *Alliance Cad System* router called *Nero*. For comparison purposes, the double patterning friendly detailed routing algorithm presented in [8] was implemented, and compared to that with the proposed algorithm OCRA. Table II provides the results of that comparison in terms of run-time, number of stitches and number of conflicts. It is clear that OCRA decreases the number of conflicts by 62% with a small increase in the run time.

TABLE II COMPARISON BETWEEN DPF WITH AND WITHOUT OCRA

Test Case	Nets	Routing Layers	Run-time (seconds)		Number of Stitches		Number of conflicts	
			[8]	*OCRA*	*[8]*	*OCRA*	*[8]*	*OCRA*
Test 1	180	2	1	2	7	7	4	1
Test 2	230	2	1	1	3	3	5	3
Test 3	403	2	2	2	12	11	6	2
Test 4	760	3	27	28	3	3	3	1
Ratio			**1**	**1.06**	**1**	**0.96**	**1**	**0.38**

VI. CONCLUSOIN

An enhanced double patterning friendly detailed routing algorithm assisted by an online conflict resolution algorithm (OCRA) was proposed and implemented. The new algorithm decreases the number of conflicts by 62% over the basic double patterning friendly detailed routing algorithm which shows the importance of conflict resolution early during the routing phase to improve the performance. Future work can adopt more conflict resolution methods besides node flipping.

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," In Proc. SPIE, vol. 6521, pp. 65211k-1 -65211k.12, 2007.

[2] A. B. Kahng, C.-H. Park, and H. Yao. "Layout Decomposition for Double Patterning Lithography," In Proc. ICCAD, pp. 465-472, 2008.

[3] Tsann-Bim Chiou et al., "Development of layout split algorithms and printability evaluation for double patterning technology," In Proc. SPIE, vol. 6924, pp. 69243M-1 - 69243M-10, 2008.

[4] K. Yuan, J.-S. Yang, and D. Z. Pan, "Double Patterning Layout Decomposition for Simultaneous Conflict and Stitch Minimization," *IEEE Transl. Computer-Aided Design of Integrated Circuits and Systems,* vol. 29, pp 185-196, 2010.

[5] Y. Xu, et al., "GREMA: Graph Reduction Based Mask Assignment for Double Patterning Technology," Intl. Conf. Computer-Aided Design, pp.601-606, 2009.

[6] J.-S. Yang, K. Lu, M. Cho, K. Yuan, and D. Pan, "A new graph-theoretic, multi-objective layout decomposition framework for double patterning lithography," in Design Automation Conference, pp. 637 – 644, 2010.

[7] Y. Xu, C. Chu, "A Matching Based Decomposer for Double Patterning Lithography" . In Proc. Int. Symp. on Physical Design, pp.121-126, 2010.

[8] M. Cho, Y. Ban, and D. Z. Pan, "Double Patterning Technology Friendly Detailed Routing," Proc. of IEEE/ACM Intl. Conf. Computer-Aided Design, pp. 506 –511, 2008.

[9] X. Gao; Macchiarulo, L, "Enhancing double-patterning detailed routing with lazy coloring and within-path conflict avoidance," In DATE , pp.1279-1284, 2010.

[10] Y-H Lin; Y-L Li; , "Double patterning lithography aware gridless detailed routing with innovative conflict graph,".In Design Automation Conference (DAC), pp.398-403, 2010.

[11] H.A. Darwish, H.N. Shagar, Y.A Badr,Y.H. Arafa, A.G Wassal, "A hashing mechanism for rule-based decomposition in Double Patterning Photolithography", In Microlectronics (ICM),pp. 363-366, 2010

[12] http://www-soc.lip6.fr/