Post-Routing Layer Assignment for Double Patterning

Jian Sun¹, Yinghai Lu², Hai Zhou^{1,2}, Xuan Zeng^{1*}

¹State Key Lab. of ASIC & System, Microelectronics Dept., Fudan University, China

²EECS Dept., Northwestern University, U.S.A.

Abstract—Double patterning lithography, where one-layer layout is decomposed into two masks, is believed to be inevitable for 32nm technology node of the ITRS roadmap. However, post-routing layer assignment, which decides the layout pattern on each layer, thus having great impact on double patterning related parameters, has not been explored in the merit of double patterning. In this paper, we propose a post-routing layer assignment algorithm for double patterning optimization. Our solution consists of three major phases: multi-layer assignment, single-layer double patterning, and via reduction. For phase one and three, multi-layer graph is constructed and dynamic programming is employed to solve optimization problem on this graph. In the second phase, single-layer double patterning is proved NP-hard and existing algorithm is implemented to optimize single layer double patterning problem. The proposed method is tested on CBL (Collaborative Benchmarking Laboratory) benchmarks and shows great performance. In comparison with single-layer double patterning, our method achieves 73% and 27% average reduction for unresolvable conflicts and stitches respectively, with only 9% increase of via number. When double patterning is constrained on only the bottom two metal layers as in current technology, these numbers become 62%, 8% and 0.42%.

I. INTRODUCTION

As feature sizes keep shrinking to 32nm while new lithography techniques such as extreme ultra violet and high index fluids immersion have yet to come, double patterning lithography has become one of the most promising lithography process for 32nm node and beyond. Double patterning technique strives to decompose features on critical layer of layout with spacing less than a predefined threshold into two masks. Features can be sliced for further decomposition, which results in additional cost of stitches. However, due to the high density of a layout, it is still possible that double patterning conflict cannot be resolved even with feature slicing [1]. The problem of how to minimize the number of the unresolvable double patterning conflict (URC) and stitch number has recently drawn much attention. Much work [1], [2], [3], [4] has been proposed to solve this problem. Although all of these algorithms are heuristics, the complexity of double patterning problem has not been theoretically proved. Only the authors of [2] claimed that they believe double patterning problem is NP-hard.

Kahng et al. [1] formulated the single-layer double patterning problem as an odd cycle elimination problem. In their work, odd cycles are detected after the construction of conflict graph. Then each detected cycle is eliminated by node splitting, which introduces a stitch. The proposed node splitting approach has been widely used in later work. In [4], double patterning problem is formulated as a max-cut problem. Equipped with graph size reduction techniques, the problem is solved efficiently, which makes it possible to decompose some huge layouts. In spite of the great effort of [1] and [4] on reducing URC's and stitches, there are still many conflicts that cannot be resolved in designs, especially in those with high feature density.

To further reduce the number of unresolvable conflicts, it is better to consider double patterning in early stages of IC design flow. In [2], [3], double patterning friendly detailed routing is proposed. Congested features are generally avoided in detailed routing stage, with additive double patterning penalty included in maze routing. This pioneer work focuses on the design patterns of a single layer. Following this direction and extending the design consideration to multiple layers, we propose in this work to use post-routing layer assignment to improve the quality of double patterning. Our proposal is based on two major reasons. First, post-routing layer assignment deals with multiple layers and has a good opportunity to consider the interaction among features from different layers. Secondly, the input data of post-routing layer assignment is the layout after detailed routing, which is the same as existing double

patterning techniques. Therefore, it can be effortlessly integrated into the current design flow.

In post-routing layer assignment, wire segments are re-assigned to proper metal layers to improve design parameters such as via numbers and wire capacitance. Via minimization problem has been well studied for several decades. Algorithms for both two-layer assignments [5], three-layer assignments [6], and multi-layer assignments [7] have been proposed. In [7], the multi-layer assignment problem is represented by an extended-continuation-conflict graph and a dynamic programming heuristic is used to solve the optimization problem on the graph. We leverage the similar philosophy and technique in our layer assignment algorithm. However, double patterning optimization makes our problem more challenging.

In this work, we developed a systematic post-routing layer assignment algorithm to effectively and efficiently address the double patterning problem. Our approach contains three major phases: (1) multi-layer assignment considering double patterning risk, (2) single-layer double patterning, and (3) via reduction. In phase one and three, we first construct multi-layer graph. Dynamic programming is then employed to solve optimization problem on this graph. The second phase, single-layer double patterning, is proved NP-hard. Single-layer double patterning algorithm proposed in [4] is conducted on all layers in this phase. The effectiveness and efficiency of our approach is demonstrated through experiments on several test cases of very different sizes. Compared with directly single-layer double patterning, the number of URC is decreased by 73%, and number of stitches is reduced by 27% in average, with only 9% increase of via numbers. When double patterning is constrained only on the bottom two metal layers as in current technology, these numbers change to 62%, 8% and 0.42%, which is more desired in via number critical designs. Our key contributions include:

- The NP-hardness proof of the single-layer double patterning problem, which settles an important open problem;
- The formulation of the double patterning layer assignment problem to achieve better benefits with global view;
- The three-phase algorithm that effectively and efficiently solves the double patterning layer assignment problem.

The rest of this paper is organized as follows. The motivation of our work and double patterning layer assignment problem formulation is given in Section II. Section III gives the general flow of our double patterning layer assignment algorithm, and the details of the algorithm phases are individually described in Section IV. NP-hardness of single-layer double patterning is proved in Section IV-C. Section V gives the experimental results. Section VI concludes this work.

II. MOTIVATION AND PROBLEM FORMULATION

It is well known that feature slicing cannot resolve all double patterning conflicts. As is shown in Fig.1(a), when single-layer double patterning is conducted directly on the given layout, even with feature slicing, a double patterning conflict is inevitable. In previous double patterning work, area-increasing layout adjustment is required to resolve such conflict. But actually, the conflict can be easily eliminated by exchanging some features between two metal layers with two more vias introduced, as demonstrated in Fig.1(b). The example shows the potential that, with double patterning consideration in layer assignment, some of URC can be eliminated with a few via increase. This motivates us to consider double patterning during post-routing layer assignment. Notice that although the given layout instance is bidirectionally routed, general layout also have the same chance to eliminate unresolvable conflicts by layer assignment. All the discussions and algorithms through the following sections are appropriate for general layout.

* Corresponding author. E-mail: xzeng@fudan.edu.cn

Fig. 1. Motivation of Double Patterning Layer Assignment

We formulate the double patterning layer assignment problem as follows.

Given: Post routing valid multi-layer layout; critical distance; position of via candidate insertion points; via size of each via candidate; number of layers that requires double patterning k ; minimum distance d_{dp} for double patterning conflict and minimum stitch overlap length o .

Find: Layer assignment and color assignment solution for all features to minimize the cost function,

$$\text{Min} \quad \sum (\alpha \cdot \text{URC} + \beta \cdot \text{Stitch} + \gamma \cdot \text{Via}), \quad (1)$$

where α, β, γ are weighting constants, and URC indicates UnResolvable double patterning Conflict, which is the most important factor in our problem.

Subject to: (i) The layer and color assignment are both valid (Section IV-A and IV-B) for each feature; (ii) No open or short circuit is induced during the assignment (If that is violated, big penalty number is added into the objective function. Section IV-A).

III. GENERAL IDEA AND FRAMEWORK OVERVIEW

Even though layer assignment and double patterning have been studied intensively, the double patterning layer assignment problem has not been attacked. Since the layer assignment is known to be NP-hard [6], and we proved the double patterning problem is also NP-hard in Section IV-C, it is intractable to exactly solve it in polynomial time. A good heuristic algorithm will thus be developed.

In previous double patterning studies [1], [4], a double patterning conflict graph is constructed for each layer of layout. Rectangles on the mask are represented by graph nodes, and relationships among them are denoted with double patterning conflict edges and touching edges. Node bi-coloring solution is then found for the graph to maximize the number of differently colored conflicting edges and uniformly colored touching edges.

In traditional multi-layer via minimization layer assignment problem [7], the extended-continuation-conflict graph is constructed from the multi-layer layout, where wire segments and via candidates are represented by graph nodes and two kinds of edges, conflict edges and continuation edges, exist in the graph. Subsequent optimization program tries to find layer assignments for all nodes to minimize the number of vias without inducing open or short circuit.

We find that the conflict graph in double patterning and the extended-continuation-conflict graph in layer assignment have similar structures,

which makes it possible to combine them together to solve our multi-layer double patterning layer assignment problem. Based on this observation, we develop the Double Patterning Layer Assignment algorithm (DPLA). In DPLA, the double patterning conflict edge in conflict graph of single-layer double patterning is referred as *Double Patterning Risk*. As shown in Fig. 2, the DPLA algorithm consists three phases:

- Multi-layer assignment, where double patterning risks and vias are minimized simultaneously;
- Single-layer double patterning, where unresolvable conflicts and stitches on each layer are minimized;
- Via reduction, where vias number is reduced without increasing unresolvable conflicts.

Fig. 2. Overall DPLA Flow

In phase 1, given a valid multi-layer layout, a graph is first constructed to represent all types of relationships among features on all layers. Dynamic programming technique is then employed to solve this optimization problem on iteratively generated induced trees. At the end of these operations, the summation of the numbers of vias and double patterning conflicts is minimized.

In phase 2, double patterning layout decomposition is operated on each of the layers generated in phase 1. We implement the algorithm proposed in [4] for this phase. Due to the double patterning conflict minimization considered in phase 1, the layout decomposition encounters less URC's and stitches.

The first two phases mainly target at the conflicts and the stitches but not the vias. Because of that reason, there are cases where the numbers of URC's and stitches are reduced at a large expenses of extra vias. Redundant vias might be introduced to reduce resolvable conflicts, which makes room for further optimization. Phase 3 is employed to further reduce the redundant vias introduced. In this phase, the graph constructed in phase 1 is updated with coloring information coded on each node (Section IV-B). Based on the new graph, a dynamic programming heuristic is developed to improve the quality of the

solution.

IV. LAYER ASSIGNMENT FOR DOUBLE PATTERNING

In this section, we present the detailed algorithms in the first and third phases of our layer assignment for double patterning. We also provide a proof that the single-layer double patterning problem is NP-hard, where we use an existing heuristic [4] to solve the problem.

A. Multi-Layer Assignment

The first phase is an effective multi-layer assignment method to minimize the numbers of both double patterning risks and vias.

Given a post-routing multi-layer layout, the wires are partitioned into a bunch of wire segments based on via candidate insertion positions. Each of the wire segments is mapped to a graph node (v_{seg}). At each end of every wire segment, a via node (v_{via}) is added to the graph node set. According to the electrical connection relationship, continuation edges (e_{con}) are introduced between segments of the same net. A segment is always connected to its via nodes. Two electrically connected v_{seg} 's are separated by a via candidate (v_{via}), and therefore connection like $v_{seg}-v_{seg}-e_{con}$ does not exist. $v_{via}-v_{via}-e_{con}$ does not exist, either, because two v_{via} 's should be merged into one if they are electrically connected. After continuation edges are connected, crossing edges (e_X) and double patterning edges (e_{dp}) are also inserted into the graph. For each node, the distances from itself to all of its neighboring non- e_{con} -connected nodes from different nets are measured. If the distance between two nodes is equal to or less than critical distance, e_X is added for this node pair. If the distance is greater than critical distance but smaller than the length of double patterning conflict minimum distance d_{dp} , e_{dp} connects them then. e_X and e_{dp} exist between $v_{seg}-v_{via}$, $v_{seg}-v_{seg}$ and $v_{via}-v_{via}$. The meaning of the three kinds of edges can be denoted like this:

- e_{con} : node-pair it connects has to be assigned to connected layers in the resultant circuit, otherwise, open circuit is introduced.
- e_X : node-pair it connects cannot be assigned to connected layers, otherwise, short circuit is introduced.
- e_{dp} : node-pair it connects should be assigned to unconnected layers as much as possible to minimize the double patterning risks between them.

“Connected layers” is used instead of “the same layer” because the layer assignment of v_{via} is not as simple as v_{seg} . The layer assignment of node is explained in the following part.

We have a decision variable l_v giving the layer assigned to each $v \in V$. The range of this variable is different for a segment and a via. It is $\{0, \dots, k-1\}$ for segments where k is the number of layers. For vias, it is $\{(0, 0), (0, 1), \dots, (k-2, k-1), (k-1, k-1)\}$, which is a set of integer pairs. It is inconvenient to handle these two ranges at the same time. So, for v_{via} , the integer pairs (x, y) are mapped to integers with function $x + \frac{(2k - (y - x) + 1)(y - x)}{2}$ [7]. Then both v_{seg} and v_{via} have integral range with different feasible region. They are denoted as $L_{seg} = \{0, \dots, k-1\}$ and $L_{via} = \left\{0, \dots, \frac{(k+1)k}{2} - 1\right\}$ for v_{seg} and v_{via} , respectively.

The weight of node is stored in a vector w_v for v , which denotes via height under different layer assignments for v_{via} , and is zero vector for v_{seg} .

Edges e_{con} and e_X have weight matrix $M_{(u,v)}$. The size of the matrix is determined by its two connected nodes. Each entry of the matrix stands for the electrical connection correctness of the two connected nodes with corresponding layer assignment. After the layer assignments of nodes are decoded into metal layers, whether the two connected nodes share the same metal layer can be easily checked. Recalling the introduction of e_{con}/e_X , if two connected nodes share the same metal layer, they are physically connected. The value of the entry is set to a big number MAX if open/short circuit is introduced by the corresponding layer assignment for its connected nodes. Otherwise, it is set to ZERO. This matrix is used as the penalty function for incorrectness of the resultant circuit. Similar strategy is used for e_{dp} edge. The entry of e_{dp} matrix is the number of double patterning risks of its corresponding nodes layer assignments, where double patterning risk is equivalent

to double patterning conflict edge in single-layer double patterning conflict graph. This can be better understood by the following example. Suppose there are two v_{via} 's connected by a e_{dp} , and they are assigned to (0, 2) and (1, 4) layers. Then they have double patterning conflict edges between them on layer 1 and 2. The entry corresponding to such assignment equals 2.

By now, all of the key parts of the multi-layer graph have been explained. We call this graph Multi-layer Graph, which is defined as follows.

Definition 1 (Multi-layer Graph): Multi-layer graph is a graph $G = (V, E)$, whose node set V has two exclusive subsets V_{seg} , V_{via} , and edge set E has three exclusive subsets E_{con} , E_X and E_{dp} . Each element $v \in V$ has its relevant layout shape, its layer assignment variable l_v , whose range is L_{seg} for $v \in V_{seg}$ or L_{via} for $v \in V_{via}$, and its weight vector w_v . Each element $(u, v) \in E$ has its relevant inter-shape relationship in the layout and its weight matrix $M_{(u,v)}$.

As an example, the multi-layer graph construction of layout in Fig. 1 is shown in Fig. 3.

Fig. 3. Construction of Multi-layer Graph

Based on the definition of multi-layer graph, the optimization problem in multi-layer assignment can be formulated as follows,

$$\begin{aligned} \text{Min} \quad & \alpha_{dp} \cdot \sum_{(u,v) \in E} M_{(u,v)}(l_u, l_v) + \sum_{v \in V} w_v(l_v) \\ \text{s.t.} \quad & l_v \in L_{seg}, \quad \forall v \in V_{seg} \\ & l_v \in L_{via}, \quad \forall v \in V_{via} \end{aligned} \quad (2)$$

where α_{dp} is used as the weighting constant for double patterning risk. A larger weight will reduce conflicts at the cost of introducing more vias.

This optimization problem has just the same properties as required by [7], which are,

- The costs are distributed on vertices and edges, and the total cost is the summation of weighted costs over all vertices and edges (Formula 2).
- The cost of a vertex is determined only by its layer assignment ($w_v(l_v)$).
- The cost of an edge is determined only by the layer assignment of the two vertices that it connects ($M_{(u,v)}(l_u, l_v)$).

Therefore, this multi-layer assignment can be optimally solved, if the graph is a tree, and approximately solved for general one, as [7] stated. Maximal induced subtree is first generated and then solved with dynamic programming.

After few iterations of the graph nodes traversal, Problem(2) is solved, but not our multi-layer double patterning layer assignment problem (Problem 1). The entry of the e_{dp} matrix only denotes the number of double patterning risks, which is the potential URC in the final result, because the actual double patterning conflict is not only related to the

layer assignment, but also determined by the detailed coloring. Single-layer double patterning is needed now to decide the color for all of the layout rectangles on each metal layer.

The layer assignment result will be processed in phase 2 for single layer double patterning. In our program, we implement the algorithm of [4] for phase 2. The double pattern partition resulted from phase 2 will be annotated to the multi-layer graph, and used in phase 3.

B. Via Reduction

After phases 1 and 2, there may still be extra vias that could be reduced. This is due to the fact that the objective function optimized in phase 1 is the number of double patterning risks, some of which may be resolved after the real double patterning in phase 2, making some introduced vias in phase 1 unnecessary. Fig. 4 gives such an example. The original layout in Fig.4(a) has two double patterning conflict edges and six vias. And the two conflicts can be solved easily with double patterning. However, after our multi-layer assignment operation, the layout is changed to the one in Fig.4(b), which has no double patterning conflict but two more vias. To eliminate these unnecessarily vias, another via reduction phase is required.

Fig. 4. Motivation of Via Reduction

Via reduction is also based on multi-layer graph. However, the original multi-layer in phase 1 needs to be updated due to two sources of modification. The first source is the node splitting in single-layer double patterning. Node splitting in single-layer double patterning resolves double patterning conflicts with introduction of stitches. As the consequence, $v_{seg}-v_{seg}-e_{con}$ shows up in multi-layer graph to connect the split nodes. With existence of this edge, the small segments belonging to the same original one always change layers at same time to guarantee the electrical connection. So the wires in final layout would not change layers at any other point except for the input via candidates.

The second source of update for multi-layer graph is the color assignment information on each node. Color assignment for v_{seg} is easy to understand. It is just the same as that in single-layer double patterning. Assignment for v_{via} , on the other hand, is more complicated. Its color cannot be represented by a binary integer, since the via is not only on one metal layer. Besides that, different layer assignments for a via will result in different number of metal layers that the via goes across. To solve these problems and unify color assignment for v_{seg} and v_{via} , the color information on all metal layers of a node is encoded as an unsigned integer number c_v , each two bits of which stand for a metal layer. For each metal layer, the color of the node can be 0 ($c_v = 00$), 1 ($c_v = 01$) or undecided ($c_v = 1X$). Note that only the colors corresponding to the current layer assignment of a node (l_v) are useful and need to be determined. For example, considering a via node v_{via} has (0, 1) as its current layer assignment in a 4 double patterning layers condition, the following two binary integers means the same color information for v_{via} , 00000101 and 11110101. Only the 0, 1 metal layer related bits, the lower 4 bits, are useful under current layer assignment.

Taking color into consideration, the weight on e_{dp} , (u, v) , can be now represented by a 4D matrix $F_{(u,v)}(l_u, l_v, c_u, c_v)$ to record the cost of URC. The entry in $F_{(u,v)}(l_u, l_v, c_u, c_v)$ denotes the number of metal

layers, where u, v are same colored, under layer assignment l_u, l_v and color assignment c_u, c_v . As the three properties required in Section IV-A are satisfied, theoretically speaking, it can be solved using the similar way. However, due to the complexity of color information added on each node, if the original algorithm is straightforward applied, the computational cost would be much higher. During the original dynamic programming process, for v_{seg} , k different layers are evaluated, and among which the optimal one is chosen. Now, this number changes to $2k$ for two colors on each metal layer. Though it is increased, it is still $O(k)$. For v_{via} , on the other hand, the situation is much worse. This number changes from $O(k^2)$ to $O(k^2 2^k)$, because a via of height k has 2^k different color combinations. Similar situation happens on e_{con} edges, since $F_{(u,v)}(l_u, l_v, c_u, c_v)$ is also used for e_{con} , instead of $M_{(u,v)}(l_u, l_v)$, to record the cost of stitches.

To overcome newly introduced complexity problem of color assignment, we propose the incremental coloring technique. During the layer re-assignment in via reduction phase 3, if any metal layer of a node in its new layer assignment has not been colored before, it is free to choose one from the two colors for this metal layer to make it introduce less URC. If the metal layer had been colored, its color will not be changed any more. This consideration can be better explained with an example as follows. Assuming that a v_{via} , u , is assigned at (2,3) before via reduction phase, u has only been colored on metal layers 2 and 3 in phase two. If the layer assignment of u is changed to (1,3), the color on its metal layer 1 can be either 0 or 1, but metal layer 2, 3 may not change any more. $F_{(u,v)}(l_u, l_v, c_u, c_v)$ can be substituted with $M_{(u,v)}(l_u, l_v)$ for deterministic color assignment of u and v then. So the possible options in dynamic programming for each node remains the same as in Section IV-A. This strategy is reasonable, because of the solution obtained from the single layer double patterning phase should not be changed in via reduction phase.

1. Put all nodes in *UNVISITED*
2. **while** *UNVISITED* $\neq \emptyset$
3. Randomly choose r from *UNVISITED* as root
4. Growing a none- e_{dp} -connected induced *TREE* from r
5. Delete all *TREE* nodes from *UNVISITED*
6. **for** each v in backward traversal of *TREE*
7. **for** each layer assignment $l_v \in L_v$
8. **if** l_v contains uncolored metal layer in c_v
9. Update c_v on uncolored layer to minimize URC
10. $w'_v[l_v] = w_v[l_v] + \sum_{\substack{(v,u) \in E \\ u \notin TREE}} M_{(v,u)}[l_v, l_u]$
11. **for** each u that $parent[u] = v$
12. $m_u[l_v] = \{l_u \mid \min_{l_v \in L_v} \{M_{(v,u)}[l_v, l_u] + w'_u[l_u]\}\}$
13. $w'_v[l_v] = w'_v[l_v] + M_{(v,u)}[l_v, m_u[l_v]] + w'_u[m_u[l_v]]$
14. $l_r = \{i \mid \min_{i \in L_r} w'_r[i]\}$
15. **for** each v in forward traversal of *TREE*
16. $l_v = m_v[l_{parent[v]}]$
17. **if** $(\alpha \cdot URC + \beta \cdot stitch + \gamma \cdot via)$ reduced **then goto** 1.

Fig. 5. The Algorithm for Via Reduction

The goal of this phase is changed, then, to minimizing the sum of stitches and vias without increasing the number of URC. To achieve this objective, each pair of nodes on the same induced tree cannot be connected with e_{dp} . Because incremental coloring strategy requires that all of u 's e_{dp} connected nodes have determined layer assignment when assign color for u to minimize the number of URC. However, the layer assignments of tree nodes are not decided until the end of dynamic programming. So, e_{dp} connected nodes with the tree nodes have to be excluded during construction of induced tree. The algorithm is accordingly modified as shown in Fig. 5. The objective function is iteratively improved in this algorithm. In each iteration (line 2-16), several induced subtrees are constructed (line 3-5) and dynamic programmed (line 6-16) one by one. During the dynamic programming, color information is kept updated as line 9.

C. NP-hardness of Single-Layer Double Patterning

Previous algorithms solving single-layer double patterning problem, including [4] we implemented in phase 2, are all heuristics. However,

NP-hardness proof of single-layer double patterning problem is still an open problem. We prove single-layer double patterning problem NP-hard as follows.

Given a layout, minimum double patterning conflict distance and overlap margin, a graph can be constructed with double patterning conflict edges and touching edges [1]. Each edge has weight that represents effort needed to overcome the double patterning conflict or handle the stitch. A bi-coloring solution is needed to maximize/minimize the differently colored conflict/touching edge connecting node pairs. With $c(u)$ denoting color of node u , positive $w(u, v)$ denoting weight of edge (u, v) and C, T representing conflict edge set and touching edge set, single-layer double patterning problem (DPP) may be stated as

$$\text{Max} \sum_{\substack{(u,v) \in C \\ c(u) \neq c(v)}} w(u, v), \quad \text{Min} \sum_{\substack{(u,v) \in T \\ c(u) \neq c(v)}} w(u, v)$$

The decision version of DPP is: Given conflict graph and threshold th_c and th_t , decide whether there is coloring solution that makes

$$\sum_{\substack{(u,v) \in C \\ c(u) \neq c(v)}} w(u, v) \geq th_c, \quad \sum_{\substack{(u,v) \in T \\ c(u) \neq c(v)}} w(u, v) \leq th_t,$$

The decision version of DPP is defined as Sub-DPP (SDPP).

To prove SDPP is NP-complete, a proved NP-complete problem is needed, from which SDPP can be transformed with polynomial reduction. 4-degree Max-Cut problem (4MC) is used as the known NP problem here. The decision version of 4-degree Max-Cut problem (S4MC) can be stated as: In a undirected graph, all of whose nodes have not more than 4 degrees, find a cut whose size is not smaller than th_{mc} .

Theorem 1: Double patterning problem is NP-hard.

Proof: NP part: It is easy to see that SDPP $\in$ NP, since one can easily tell whether the two required thresholds are satisfied with some given coloring solution in polynomial time.

NP-complete part: Every graph G of maximum degree 4 has a standard mesh drawing [8]. In this drawing, nodes are assigned to different mesh points called *vertices*, edges are assigned on the mesh lines called *curves* without going through any vertices and with only finite intersection with other curves. The intersection between any two different curves is called *cross point*. Besides that, the distance between any cross point pair, vertex pair, and between any vertex and cross point are all at least 10 grids.

To make it proper for our transformation, cross points in standard mesh drawing are modified as Fig.6 shows. It is obvious that we can always find a modified mesh drawing that has all the properties a standard mesh drawing has, since the mesh drawing can be scaled up to prevent any possible property violation.

Fig. 6. Modification on Standard Mesh Drawing

Each cross point on the modified standard mesh drawing is substituted with a gadget shown in Fig.7. α 's are sections of subdivided curves, and β 's are appended points. The black line segments α and β can be viewed as 0-width interconnection wire pieces on a mesh based layout. The red edges among them are double patterning edges with double patterning conflict distance equals to $\sqrt{2}$ grids. Weights on these edges $w(u, v) = 1$. This reduced graph is called G' . As Lemma3 in [8] states, each substituted gadget makes $MaxCut(G')$ 8 more than $MaxCut(G)$ by considering each β connected triangle makes 2 contribution to $MaxCut(G)$. Besides the gadget substitution, curve is further subdivided to make it $2i$ (even number) sections connected by $2i - 1$ double patterning

Fig. 7. Gadget Used for Substitution

edges. As stated in Formula(1) in [8], the $2i - 1$ edges on each curve makes $MaxCut(G')$ $2i - 2$ more than $MaxCut(G)$. Notice that, this kind of further subdivision has to be far from the existing gadgets to prevent introducing other double patterning edges. Because the length of the curves are long enough, this subdivision can always be found.

The transformed graph G' is actually a special case layout with 0 wire width, $\sqrt{2}$ double patterning conflict length, no touching edge and mesh based routed. If a cut on l -edge 4-degree graph G has a size not less than th_{mc} , after m gadget substitution and curve subdivision leading to $2t$ sections, a cut on G' can be easily found with not less than $th_{mc} + 8m + 2t - 2l$ size. With $th_c = th_{mc} + 8m + 2t - 2l$, the special case of SDPP has the same solvability as S4MC.

The reduction from S4MC to the special case of SDPP is obvious polynomial, because the construction and modification of standard mesh drawing, gadget substitution and curve subdivision are all polynomial time achievable. So the decision version of single layer double patterning problem is NP-complete. ■

It worths notice that in Fig.7 Euclidean distance is considered. If Hamilton distance is considered as [9], the two crossing red edges in the center of the gadget do not exist. The constructed graph becomes planar, where MaxCut is polynomial time solvable. The double patterning problem is not NP-hard when Hamilton distance is considered instead of Euclidean distance.

V. EXPERIMENTAL RESULTS

A. Experimental Setup

Our DPLA algorithm is implemented in C++. All of the experiments are performed on a Linux workstation with 3.0GHz CPU and 2.0GB memory. Since none of the previous double patterning work is tested on multi-layer interconnect layout, DPLA is tested on the benchmarks provided by the authors of [7], which are also available at the Collaborative Benchmarking Laboratory (CBL). The layouts are scaled down to 30nm for the experiment.

B. Multi-Layer Double Patterning Results

Since none of the existing work focuses on the double patterning aware layer assignment problem, we compare our solution with Single-Layer Double Patterning [4] conducted on all layers separately. Since the same single-layer double patterning method is used in phase 2 of our DPLA, this comparing strategy guarantees that the performance enhancement is caused by the proposed framework. With better single-layer double patterning implementation in phase 2, the DPLA result should also improve.

On the other hand, DPLA changes the number of vias. We compared with the via minimization result in [7] as the baseline of via number. It is desired that the output result of DPLA does not induce too many extra vias comparing with the via-minimized solution.

The detailed results are listed in Table I. The columns #Node and #Edge correspond to the number of nodes and edges in multi-layer graph. The column titled #URC (#Stitch) denotes the number of unresolvable double patterning conflicts (stitches) in the circuit. The #Via column shows the number of vias in the layout result. It is obvious that the quality of double patterning layout decomposition is significantly enhanced by DPLA. Both URC and stitches are dramatically reduced by 72.56% and 27.34% (up to 81.95% and 40.34% for the best case).

TABLE I
PERFORMANCE OF DPLA OVER BASELINE DOUBLE PATTERNING AND VIA MINIMIZATION

Cases	#Node	#Edge	Single-Layer DP[4]		Via Minimization[7]		DPLA						
			#URC	#Stitch	#Via	Time(s)	#URC		#Stitch		#Via		Time(s)
test1	4328	88708	532	290	1704	5.77	66	12.41%	173	59.66%	1898	111.38%	12.50
test2	8606	307086	1339	803	4180	21.45	421	31.44%	672	83.69%	4312	103.16%	35.74
test3	11204	472908	2013	1030	5490	46.56	601	29.86%	918	89.13%	5654	102.99%	81.91
mcc1-75	11649	332446	1602	839	4917	16.06	448	27.97%	532	63.41%	5525	112.37%	55.33
mcc2-75	67555	5350970	11669	5962	34117	469.61	4147	35.54%	4021	67.44%	40157	117.70%	2033.66
average	-	-	-	-	-	-	-	27.44%	-	72.66%	-	109.52%	-

TABLE II
PERFORMANCE OF DPLA ON TWO BOTTOM LAYERS

Cases	#Node	#Edge	Single-Layer DP[4]		Via Minimization[7]		DPLA						
			#URC	#Stitch	#Via	Time(s)	#URC		#Stitch		#Via		Time(s)
test1	3600	51750	521	261	1132	2.38	133	25.53%	225	86.21%	1144	101.06%	4.03
test2	5585	82540	789	371	1620	6.09	248	31.43%	330	88.95%	1630	100.62%	13.95
test3	7394	122132	1022	540	2208	15.42	278	27.20%	485	89.81%	2210	100.09%	31.68
mcc1-75	9323	185598	1518	710	3058	7.74	748	49.28%	703	99.01%	3067	100.29%	17.44
mcc2-75	47917	1499986	8737	3538	15761	190.81	5039	57.67%	3475	98.22%	15767	100.04%	649.17
average	-	-	-	-	-	-	-	38.22%	-	92.44%	-	100.42%	-

TABLE III
EFFECT OF VIA REDUCTION PHASE IN DPLA

Cases	Pre-Phase 3					Post-Phase 3				
	#URC	#Stitch	#Via	#Stitch + #Via		#URC	#Stitch	#Via	#Stitch + #Via	
test1	67	113	2086	2199		66	98.51%	173	1898	2071 91.72%
test2	422	644	4424	5068		421	99.76%	672	4312	4984 98.34%
test3	608	882	5776	6658		601	98.85%	918	5654	6572 98.71%
mcc1-75	458	413	6001	6414		448	97.82%	532	5525	6057 94.43%
mcc2-75	4237	2579	45237	47816		4147	97.88%	4021	40157	44178 92.39%

On the other hand, the vias number only increase by about 9.52% in average, even compared with the via minimized results. The running time of DPLA is also compared with Via Minimization[7] in column Time. It has the same order of magnitude as [7].

C. Double Patterning Constrained on Two Bottom Layers

In current technology, double patterning is usually conducted on the bottom two metal layers instead of all layers, because feature density is higher on these two layers. To demonstrate our method's effectiveness in current technology, DPLA is run on all cases with only the bottom two layers. The result is shown in Table II. Although the layer assignment is constrained on only two layers, which makes the resource on higher layers unavailable, DPLA flow still works very well. There are still 61.78% and 7.56% in URC and stitch reductions in average. Furthermore, via increment changes dramatically from 9.52% to 0.42% at the same time. This is because each time a segment reassigned to the other metal layer, via increment is at most 2, that is much smaller than unconstrained situation. This promising result shows that our DPLA is profitable for current IC designs, especially the via number critical ones. Besides that, since the constructed multi-layer graph is much simpler than layer unconstrained condition, running DPLA is faster.

D. Necessity of Via Reduction Phase

Based on our observation, the majority of the URC and stitch reduction has been achieved during the first two phases. But the output of phase 2 is still far from optimality considering cases like Fig. 4. The elimination of unnecessary introduced vias only depends on the last via reduction phase. Table III demonstrates the necessity of the via reduction phase. Differences on #URC, #stitch and #via before and after phase 3 in layer unconstrained experiments are listed for comparison. In the experiment, vias and stitches have the same importance as their weighting constants β, γ are the same. It is convenient to change the importance of these objectives by adjusting the weighting constants differently. After phase 3, the total number of vias in the layout is well reduced with relatively small increase of stitches, and the URC number also reduced simultaneously. In other words, the goal of via reduction phase, minimizing the sum of stitches and vias without increase the number of URC, has obviously been achieved in all cases.

VI. CONCLUSIONS

High performance double patterning methodology is desperately needed, since the lithography process has become more and more critical in IC manufacture. In this paper, we propose a post-routing layer

assignment algorithm DPLA to efficiently address the double patterning problem. We formulate the double patterning layer assignment problem with the introduction of multi-layer graph. A dynamic programming based three-phase approach is developed to find the global solution during the flow. Experiments on the test cases of very different sizes demonstrate the high efficiency of our approach.

VII. ACKNOWLEDGEMENT

This work was supported in part by the NSFC Research Projects 60976034 and 61076033, the National Basic Research Program of China under Grant 2005CB321701, the National Major Science and Technology Special Projects 2008ZX01035-001-06 and 2009ZX02023-4-3 of China during the 11th Five Year Plan Period, the Doctoral Program Foundation of the Ministry of Education of China 200802460068, the Program for Outstanding Academic Leader of Shanghai and NSF under CCF-0238484 and CCF-0811270.

REFERENCES

- [1] A. B. Kahng, C. H. Park, X. Xu, and H. Yao, "Layout decomposition for double patterning lithography," in *ICCAD*, 2008.
- [2] M. Cho, Y. Ban, and D. Z. Pan, "Double patterning technology friendly detailed routing," in *ICCAD*, 2008.
- [3] K. Yuan, K. Lu, and D. Z. Pan, "Double patterning lithography friendly detailed routing with redundant via consideration," in *DAC*, 2009.
- [4] Y. Xu and C. Chu, "GREMA: Graph reduction based efficient mask assignment for double patterning technology," in *ICCAD*, 2009.
- [5] R. Y. Pinter, "Optimal layer assignment for interconnect," in *Proc. IEEE Int. Conf. Circuits and Computers*, 1982, pp. 398-401.
- [6] K. C. Chang and H. C. Du, "Layer assignment problem for three-layer routing," *IEEE Trans. Comput.*, vol. 37, pp. 625-632, 1988.
- [7] C. C. Chang and J. Cong, "An efficient approach to multilayer layer assignment with an application to via minimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, pp. 608-620, 1999.
- [8] J. Diaz and M. Kaminski, "MAX-CUT and MAX-BISECTION are NP-hard on unit disk graphs," *Theoretical Computer Science*, vol. 377, pp. 271-276, 2007.
- [9] Y. Xu and C. Chu, "A matching based decomposer for double patterning lithography," in *ISPD*, 2010.