

# Flexible Self-aligned Double Patterning Aware Detailed Routing with Prescribed Layout Planning

Jhii-Rong Gao and David Z. Pan  
ECE Dept. Univ. of Texas at Austin, Austin, TX 78712  
{jrgao, dpan}@cerc.utexas.edu

## ABSTRACT

Self-aligned double patterning (SADP) is a promising manufacturing option for sub-22nm technology nodes. Studies have shown that SADP provides better overlay control than traditional litho-etch-litho-etch double patterning. However, the use of stitch is not allowed, which makes layout decomposition for SADP more difficult. It is necessary to find a new solution to handle pattern conflicts and consider SADP in earlier stages. In this paper, we propose a novel multi-layer SADP-aware detailed routing with prescribed layout planning. Our method is based on a correct-by-construction approach to take SADP compliancy into account during routing, and to achieve layout decomposition simultaneously. The experimental result shows that the proposed approach consistently achieves SADP-compliant solutions on both single-layer and multi-layer designs.

## Categories and Subject Descriptors

B.7.2 [Hardware, Integrated Circuit]: Design Aids - Placement and Routing

## General Terms

Algorithms, Design, Performance

## Keywords

Detailed Routing, Double Patterning, SADP, Design for Manufacturability

## 1. INTRODUCTION

Due to the delay in the next generation lithography technology such as Extreme Ultra Violet (EUV) [1], the manufacturing industry still relies on a 193nm (ArF) wavelength light source. As technology continues to scale to 22nm and 14nm, semiconductor manufacturing with ArF is greatly challenging because the required half pitch size is beyond the resolution limit of ArF. Double Patterning Lithography (DPL) has been a promising solution for 22nm/14nm node volume production.

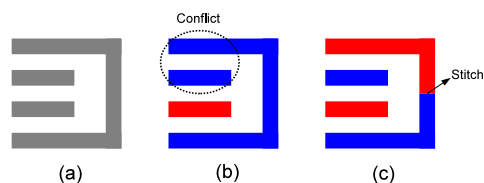
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISPD'12, March 25–28, 2012, Napa, California, USA.

Copyright 2012 ACM 978-1-4503-1167-0/12/03 ...\$10.00.

The working principal of DPL is to decompose dense layout patterns into two masks. The decomposition process is referred to as coloring. Since each mask contains sparse patterns with doubled spacing, the lithography resolution can be improved. There are two main DPL schemes in current IC manufacturing: litho-etch-litho-etch (LELE) double patterning, and self-aligned double patterning (SADP). LELE consists of two exposure and two etch processes [2–4], and it allows stitch insertion to resolve the conflict after pattern decomposition. Stitch is used to split a pattern into two masks, which makes patterns even sensitive to process variation. Fig. 1 shows how stitch insertion resolves conflicts. Possible variations may occur at the stitch inserting point, such as line-end shortening, CD shrinking [4], and overlay error due to two consecutive mask exposure processes. Several layout decomposition methods [3–6] have been proposed for LELE DPL. However, the alignment and magnification errors on the second mask exposure cause LELE to induce significant pattern overlay error [7] and thus degrade yield rate.

SADP is similar to LELE, which also requires layout decomposition into two masks, core mask and trim mask. For the mandrel pattern that is lithographically defined on the core mask, sidewall spacers are applied on each side to effectively double the pattern density. The trim mask is then used to remove unnecessary patterns. Because the most critical patterning control in SADP is not governed by lithography, but by the deposition of the sidewall spacer, it has less overlay error and excellent variability control compared to LELE [8]. However there is no stitch allowed in SADP to resolve conflicts, making its layout decomposition difficult. Moreover, SADP layout decomposition is not intuitive in the sense that the decomposition result does not have a direct relation to the original layout. SADP requires assist mandrels [9] during its patterning process and these unwanted mandrels need to be trimmed out by the trim mask. There-



**Figure 1: Conflict and stitch in LELE. (a) Target layout. (b) A conflict occurs after layout decomposition. (c) The conflict is resolved by splitting a pattern with stitch insertion.**

fore, the core mask and the trim mask cannot be obtained simply from the target layout. For 2D patterns, this mask assignment process would be more complicated.

Recently, SADP has attracted more and more interest because of its good overlay controllability. Starting from 1D pattern decomposition [10], the flexibility of SADP is further extended to 2D random logic patterns [11, 12]. The layout decomposition problem is solved with an ILP formulation [13] to minimize overlay. A lithography friendly algorithm to automatically generate core pattern, trim patterns, and assist core patterns for better manufacturability is proposed in [9].

Most existing SADP-related works focus on post layout optimization. As described above, the prohibition against using stitch makes SADP decomposition extremely challenging. Therefore, it is necessary to consider SADP in earlier design stages, especially in detailed routing, to guarantee high layout decomposability. Double patterning friendly routing has been proposed in [14, 15], but their methods cannot be applied to solve SADP induced issues. A SADP-friendly detailed routing flow [16] is presented by performing detailed routing and layout decomposition concurrently. However, [16] simply works for single-layer designs and does not provide a solution for resolving layout decomposition conflicts.

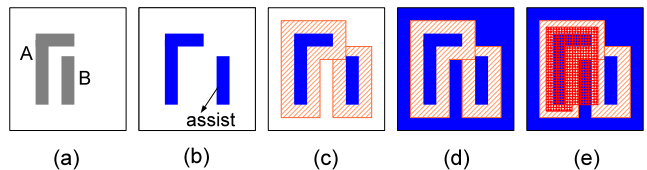
In this paper we propose a robust multi-layer SADP-aware detailed routing algorithm which includes the following features:

- We propose a novel SADP-aware detailed routing approach that can handle 2D patterns on multi-layer designs in the presence of obstacles.
- We solve routing and layout decomposition simultaneously based on the correct-by-construction approach.
- We incorporate layer assignment to resolve potential pattern conflicts, which increases the flexibility of layout decomposition for SADP.
- We present a set of SADP-aware routing guidelines, which helps improve the pattern quality of SADP.

The rest of the paper is organized as follows. Section 2 gives preliminary information on SADP technology and our work. Our prescribed layout planning techniques are explained in Section 3. The details of the proposed SADP-aware routing framework are presented in Section 4. The experimental results are discussed in Section 5, followed by the conclusion in Section 6.

## 2. PRELIMINARIES

Two main process flows are available for SADP, positive tone process and negative tone process. These two processes are analyzed in [8] which suggested positive tone process is preferred due to its more cost effective and better controllability of overlay. Fig. 2 shows the process flow of the positive tone SADP. Assume (a) is the target layout, it can be generated by the process from (b) - (e). First of all, the core mask shown in blue in (b) is derived by selecting a subset of target layout patterns and assist patterns. The patterns on the core mask are called main mandrel and are generated through a lithography process. Then the sidewall spacers are deposited into both sides of the main mandrel as shown



**Figure 2:** Flow of positive tone SADP. (a) Target layout. (b) Core mask. (c) Spacer deposition. (d) Substrate material filling. (e) Trim mask.

in (c). Next, main mandrels are removed and the non-spacer region is filled with substrate materials. In the final step (e), the trim mask shown in red is applied to remove unnecessary patterns and retain the desired patterns.

Similar to LELE, SADP requires layout decomposition to separate patterns into two subsets. The decomposition result is usually obtained by performing coloring on all patterns. One subset can be generated by directly assign the patterns into core mask. While the other has to be generated by forming a trim mask which can retain the patterns on the final layout. In this paper, we define patterns that are printed by core mask as *mandrel patterns*; and patterns that are printed by the assistant of trim mask as *trim patterns*. A route can be assigned as either mandrel pattern or trim pattern. For example, Fig. 2(e) shows the case when pattern *A* in (a) is assigned as mandrel pattern and *B* as trim pattern.

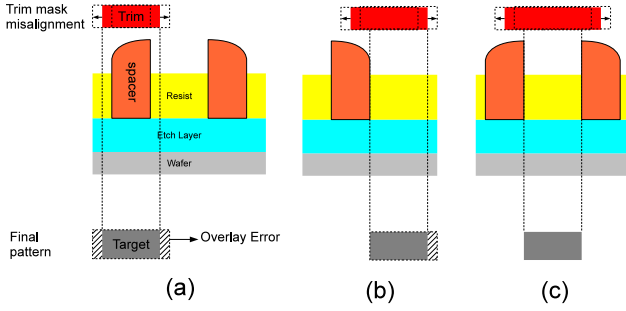
Our approach adopts the grid-based routing model. Because multi-layer designs are taken into consideration, a three-dimensional grid graph is constructed. Our routing not only searches solutions on single metal layer, but also allows solutions crossing multiple layers through vias. Each pin is mapped to one grid, and a routing solution of a multi-pin net is composed of grids connecting all of its pins. Here we define the minimum width of mandrel pattern and trim pattern as  $W_{min}$ . We also assume that the width of sidewall spacer equals to  $W_{min}$ . A minimum spacing  $S_{min}$  must be kept between any neighboring routing patterns. For a legal layout decomposition result, patterns within the minimum DPL spacing  $S_{dp}$  must be assigned to different types of pattern (mandrel pattern or trim pattern).

## 3. PRESCRIBED LAYOUT PLANNING FOR SADP COMPLIANCY

Our objective is to achieve better SADP compliancy by performing routing and SADP layout decomposition simultaneously. As a result, the routing solutions are able to take advantage of SADP's good overlay control. In this section, we present SADP-friendly routing guidelines to improve pattern quality and reduce decomposition conflicts.

### 3.1 SADP-aware routing guidelines

Mandrel patterns and trim patterns are fabricated by different manufacturing processes. The interaction between these two types of patterns may affect the printing images. Therefore, simply determining whether a layout is decomposable is not adequate for SADP-friendly routing. We analyze the impact of different pattern assignments on the pattern quality. The following three layout planning guidelines provide a systematic procedure to construct a SADP-friendly routing. Incorporating these guideline into our rout-



**Figure 3: Overlay error due to trim mask misalignment.** (a) No feature boundary aligned to spacer. (b) One feature boundary aligned to spacer. (c) Both feature boundaries aligned to spacer.

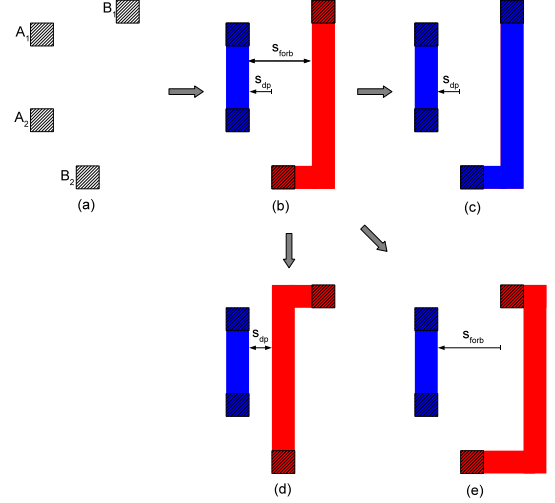
ing framework enables us to take advantage of SADP technology.

1. If both mandrel pattern and trim pattern are conflict-free when being assigned to a route, the mandrel pattern is preferred.
2. If the candidate routes have the same routing cost and can only be assigned as trim patterns, the route with more spacer protection is preferred.
3. The distance between a trim pattern and a mandrel pattern is suggested to be larger than the forbidden spacing  $S_{forb}$ ; although a valid routing solution only requires the minimum spacing  $S_{dp} < S_{forb}$  to be satisfied.

These guidelines are explained below. The simulation result in [17] observes the printability degradation for the second mask lithography due to the presence of topography generated from the first mask on the wafer. One degradation can be seen from the CD variation, where patterns from the second lithography tend to have wider width when there are underlying patterns from the first litho/etch step. As a result, SADP prefers mandrel patterns from the first lithography for better printability control, and is different from LELE which prefers two balanced subsets of patterns [4].

Another advantage of SADP is the use of spacer. Three decomposition cases showing how a trim mask can be formed [8] are presented in Fig. 3. The grey rectangle represents the target pattern that will be generated by the trim mask shown in red. In (a), a wide line is formed by the trim mask overlapping both sides of a spacer. Consequently, the trim mask misalignment may affect both the left and right boundary of the printing image. The possible overlay errors on the final pattern are shown in the slash area. In (b), a line is formed by the trim mask side-lapped with one spacer, causing overlay errors only on one side of the printing image. If both sides of the target pattern are aligned to spacers, as shown in (c), the overlay error can be totally avoided. Given this auto-alignment property of the spacer, a trim pattern protected by multiple spacers is preferred.

The minimal spacing  $S_{dp}$  in DPL constraints the minimum allowable distance between any two identical type patterns. A conflict occurs if two patterns within  $S_{dp}$  are assigned to the same mask. In addition, a forbidden spacing needs to be considered. The simulation results in [16] show that the printed image of a trim pattern would be affected by a close



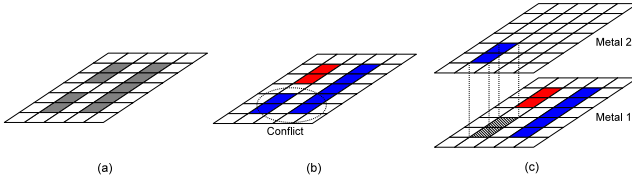
**Figure 4: Prescribed layout planning.** (a) Unrouted nets. (b) Legal patterns with bad quality. (c) - (e) Improved patterns by our prescribed layout planning.

mandrel pattern even if  $S_{dp}$  is satisfied. In contrast, the quality of a trim pattern can be improved if its neighboring mandrel patterns are kept at a sufficient distance. Therefore, we define a forbidden spacing  $S_{forb} > S_{dp}$  such that any distance  $d_{mt} < S_{forb}$  is discouraged, where  $d_{mt}$  denotes the distance between a neighboring trim and mandrel pattern.

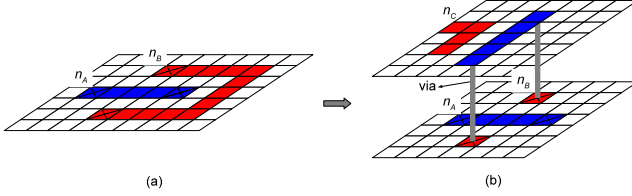
These layout planning techniques work as prescriptions for our routing engine to generate SADP-compliant layout patterns and to prevent patterns with bad quality. The example in Fig. 4 shows how routing patterns can be improved by our approach. The pin locations are given in (a) for two unrouted nets, and (b) is one routing and layout decomposition solution without considering SADP. Mandrel pattern is shown in blue and trim pattern is shown in red in our following explanation. Although (b) is a legal solution by satisfying  $S_{dp}$  constraint, the mandrel pattern and the trim pattern may affect each other because their distance are within  $S_{forb}$ . Three alternative solutions with better pattern quality are shown in (c) - (e); where (c) adopts more mandrel patterns; (d) acquires more spacer protection; and (e) enlarges the distance between neighboring mandrel and trim patterns.

### 3.2 Simultaneous layer assignment for conflict prevention

The biggest challenge of SADP is the prohibition against using stitches. For a route  $path_1$  on a single layer, either all grids in  $path_1$  are assigned as mandrel patterns or all are assigned as trim patterns. This limitation dramatically decreases the possibility of generating a decomposable layout for SADP. In order to increase the flexibility of SADP layout decomposition, we perform simultaneous layer assignment during routing. In contrast to single-layer layout decomposition, multi-layer layout decomposition allows patterns to be assigned independently if they are on different layers. For example, a route  $path_2$  is composed of  $seg_1$ - $via_{12}$ - $seg_2$ , where  $seg_1$  is on metal 1,  $seg_2$  is on metal 2, and  $via_{12}$  is used to connect  $seg_1$  and  $seg_2$ . Since  $seg_1$  and  $seg_2$  are on different metal layers, they can be decomposed independently without introducing any conflict. Via can be viewed



**Figure 5:** Prevent conflicts by simultaneous layer assignment. (a) Target layout. (b) Conflict occurs in single-layer layout decomposition. (c) Conflict removed by proper layer assignment.



**Figure 6:** Increase flexibility of layout decomposition by simultaneous layer assignment. (a) Single layer. (b) Multiple layers.

as a splitting point similar to the function of stitch in LELE layout decomposition.

Performing layer assignment during layout decomposition on multi-layer designs has two advantages for SADP compliance. First, a conflict can be easily resolved by assigning conflicting patterns into different metal layers. Fig. 5 shows a conflict that is solvable by our simultaneous layer assignment. Fig. 5(a) is the target layout that needs to be printed by DPL. A conflict occurs after single-layer layout decomposition in (b). By properly assign the patterns to different layers as shown in (c), the conflict can be prevented. The second advantage of considering layer assignment is that it increases the flexibility of layout decomposition. Fig. 6(a) shows an example after routing and layout decomposition on a single layer. Net  $n_B$  needs to detour to prevent intersecting net  $n_A$ . By assigning a section of patterns on  $n_B$  to an upper layer as shown in (b), wirelength is reduced. Besides, the patterns on different layers are not restricted to a single color. In Fig. 6(b), patterns of  $n_B$  on metal 1 is assigned as trim patterns (shown in red); while the pattern on metal 2 is assigned as mandrel pattern (shown in blue) to provide spacer protection for the routed net  $n_C$  and to prevent conflicts.

The simultaneous layer assignment technique increases the solution space of both routing and layout decomposition, and thus helps prevent conflicts. This layer assignment is integrated into our three-dimensional path finding process, which will be explained in the next section.

## 4. MULTI-LAYER SADP-AWARE DETAILED ROUTING

This section gives the detail of our proposed routing framework. We first introduce the overall flow, and then present the techniques incorporated in the flow.

### 4.1 Overall flow

We adopt a correct-by-construction approach to build our routing flow. When a net is routed, its layout decomposition

---

### Algorithm 1 SADP-aware detailed routing

---

**Input:** A set of blockages  $B$ , and a set of nets  $N$

```

1: Layout decomposition for  $B$ 
2:  $Q \leftarrow$  An arbitrary net  $n_{begin} \in N$ 
3: while  $!Q.empty()$  do
4:    $n \leftarrow Q.pop()$ 
5:   for each 2-pin net  $k \in n$  do
6:     Three-dimensional A* search for  $k$ 
7:   end for
8:   for each  $n_{neighbor} \in N$  where bbox of  $n_{neighbor}$  overlaps bbox of  $n$  do
9:      $Q \leftarrow Q + n_{neighbor}$ 
10:  end for
11: end while

```

---

**Figure 7:** SADP-aware detailed routing.

is done simultaneously. During path finding, a rule checking procedure ensure not only a route is legal but also its patterns are decomposable. Consequently, once the routing is done, its layout decomposition result is also obtained.

Algorithm 1 in Fig. 7 describes the overall flow of our approach. First, we perform initial layout decomposition for the blockages composed of pre-routed nets. Since pre-routed nets in this stage are usually sparse, most would be assigned as mandrel patterns according to Guideline 1. Next, we process the input nets sequentially according to the routing order determined in line 8-9 (Section 4.3). Each multiple-pin net is decomposed into 2-pin nets and then routed using our three-dimensional A\* search in line 5-7 (Section 4.4). The routing cost in A\* search is a combination of wirelength and SADP cost, which will be illustrated in Section 4.2. After the A\* search, the pattern assignment with lowest cost will be chosen.

### 4.2 SADP-aware weighted cost

When performing A\* search, the cost of routing on a grid is calculated. Suppose an edge connecting grid  $g_i$  to  $g_j$  is considered, the cost of routing  $g_j$  as a mandrel and as a trim pattern is defined as follows:

$$\begin{cases} cost_j(m) &= cost_i(m) + \alpha \cdot WL_{ij} + \beta \cdot SADPC_j(m) \\ cost_j(t) &= cost_i(t) + \alpha \cdot WL_{ij} + \beta \cdot SADPC_j(t) \end{cases} \quad (1)$$

if  $g_i$  and  $g_j$  are on the same layer.

$$\begin{cases} cost_j(m) &= \min \{ cost_i(m), cost_i(t) \} + \\ &\quad \alpha \cdot WL_{ij} + \gamma \cdot VIA + \beta \cdot SADPC_j(m) \\ cost_j(t) &= \min \{ cost_i(m), cost_i(t) \} + \\ &\quad \alpha \cdot WL_{ij} + \gamma \cdot VIA + \beta \cdot SADPC_j(t) \end{cases} \quad (2)$$

if  $g_i$  and  $g_j$  are on different layers.

The pre-calculated cost  $cost_i(m)$  and  $cost_i(t)$  represent the cost when  $g_i$  is assigned as a mandrel pattern and a trim pattern, respectively;  $WL_{ij}$  is the wirelength between neighboring grids  $g_i$  and  $g_j$ ; VIA is the via cost and  $SADPC$  can be either positive or negative to represent a bad or good impact on pattern quality, respectively. User-defined parameters  $\alpha$ ,  $\beta$  and  $\gamma$  adjust the weight between wirelength and SADP awareness. As mentioned previously, stitch is not allowed in SADP. Therefore,  $g_j$  must be assigned as the same pattern of  $g_i$  if they are on the same layer, just as defined in Equation 1. When multi-layer designs are involved,

more optimization options are available. Therefore Equation 2 provides more solution space when searching on multiple layers.

*SADPC* is the double patterning cost when a grid is assigned as a mandrel/trim pattern and is determined by the guidelines provided in Section 3.1, which is defined as follows:

$$SADPC = \begin{cases} C_{mandrel} & - m \cdot C_{spr} + n \cdot C_{forb} \\ C_{trim} & \end{cases} \quad (3)$$

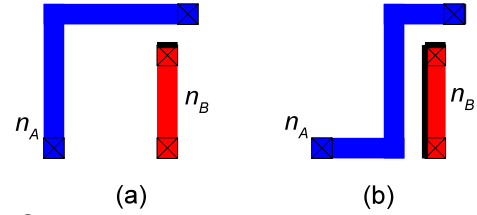
$C_{mandrel}$  and  $C_{trim}$  are the unit cost of assigning a grid as a mandrel or a trim pattern, respectively. The weight of  $C_{mandrel}$  is set to be less than the weight of  $C_{trim}$  according to Guideline 1 such that more mandrel patterns will be used.  $C_{spr}$  represents the benefit of a self-aligned spacer and thus it reduces the total *SADPC* according to Guideline 2. The number of newly generated spacer-protected grids  $m$  can be optimized by routing more mandrel patterns next to existing trim patterns, or routing more trim patterns next to existing mandrel patterns.  $C_{forb}$  represents the penalty for patterns violating  $W_{forb}$  according to Guideline 3. Similar to  $m$ ,  $n$  is the total number of newly generated forbidden grids by the current routing path. Note that violating  $W_{forb}$  is not encouraged, but it is valid for double patterning.

In general, the weight of these SADP costs differs depending on the technology. However, we may adjust the weight according to the routing density. For example, a larger  $C_{spr}$  encourages the binding of mandrel and trim patterns, and thus helps generate a tighter layout. In contrast, larger  $C_{forb}$  encourages a detour to prevent violating forbidden spacing, and thus consumes more routing resources. In our experiments, we set  $C_{mandrel}=C_{spr}=C_{forb}$  and  $C_{trim}=2C_{mandrel}$ .

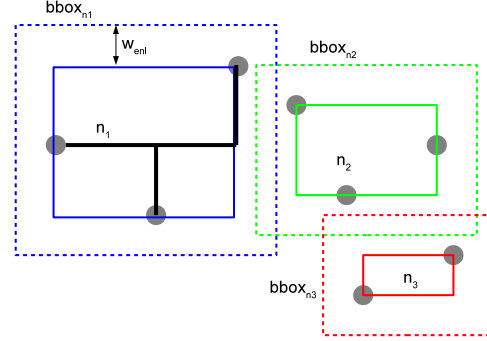
### 4.3 Neighborhood-based net ordering

How a routing algorithm explores its solution defines how important net ordering is. For an ILP-based algorithm, solutions are calculated currently, thus net ordering is unnecessary. However, ILP-based algorithms usually have high runtime overhead. On the other hand, a sequential routing algorithm that processes nets one by one relies on a good net ordering method. The better the net ordering is, the less rip-up and reroute are required and the less the runtime is needed. According to the cost function defined in Section 4.2, a preferred routing path should keep a low wirelength and has more spacer-protected grids. Fig. 8 shows the comparison of a bad and a good net ordering. In (a), net  $n_A$  is routed first and then  $n_B$  is routed. The bold line in the grid boundary shows where the grid boundary is protected. The net order of Fig. 8(b) is contrary to (a). We can see that with the same wirelength, the solution in (b) obtains much more spacer protection.

To achieve SADP-friendly net ordering, we propose an ordering method based on the geographic relation among nets. First, an arbitrary net  $n_i$  is selected to be routed. After  $n_i$  is routed, we obtain the next net to be routed  $n_j$  by finding every  $bbox_{n_j}$  overlapping  $bbox_{n_i}$ . Here  $bbox_n$  is determined by enlarging the net bounding box by a specific width  $w_{enl}$ . This ordering method encourages nets within a certain distance to be routed in a sequence, so that the probability to provide spacer protection for these neighboring nets can be increased. In our implementation, we set  $w_{enl}$  slightly larger than  $S_{forb}$  so that the enlarged area is sufficient but not causes too much computational burden.



**Figure 8: Net ordering impact on pattern quality.** Bolder lines show grid boundaries that are protected by spacers. (a) Net  $n_a$  is routed first. (b) Net  $n_b$  is routed first.



**Figure 9: Neighborhood-based net ordering.**  $n_2$  allows more spacer protection to be provided for  $n_1$ .

Fig. 9 shows an example of neighborhood-based net ordering. In the beginning, net  $n_1$  is routed and the next routing net will be determined. It can be seen that  $bbox_{n_2}$  overlaps  $bbox_{n_1}$  and thus  $n_2$  will be routed next. Finally,  $n_3$  will be routed because its  $bbox_{n_3}$  overlaps  $bbox_{n_2}$ .

Because the searching for overlapping  $bbox$  needs to be done whenever a net is routed, it is important to reduce the overhead of this search. We adopt R-tree [18] for fast indexing  $bbox$  information.

### 4.4 Efficient three-dimensional path finding by dynamic programming

During path finding, when a routing grid  $g$  is considered, the validity of assigning  $g$  as a mandrel pattern (blue) and a trim pattern (red) is checked simultaneously. The combined routing and layout decomposition result is denoted as  $R(path, LD(path))$ , where  $path$  is the routing path composed of grids, and  $LD(path)$  is the coloring result for  $path$ . If a solution candidate  $R(path_1, LD(path_1))$  generates any conflict, a high routing cost defined in Section 4.2 would be applied to prevent this candidate being selected.

The solution space for  $R(path_i, LD(path_i)) \forall i$  in single-layer SADP is limited because all grids  $g_j \in path_i$  must be assigned as the same color. However, the solution space on multi-layer designs would be much larger. As discussed in Section 3.2, simultaneous layer assignment with routing enables more flexible layout decomposition. Therefore, we adopt a three-dimensional path finding so that layer assignment can be integrated into the routing process. Fig. 10 shows a routing path connecting pins  $p_1$  and  $p_2$ . Because the path is composed of three independent segments,  $seg_1$ ,  $seg_2$ ,  $seg_3$ , which are connected by vias, each segment is flexible to be assigned as either a mandrel or a trim pattern. It can be seen that in total 8 candidate solutions are available for the case in Fig. 10.

The time and space complexity would be an issue if we

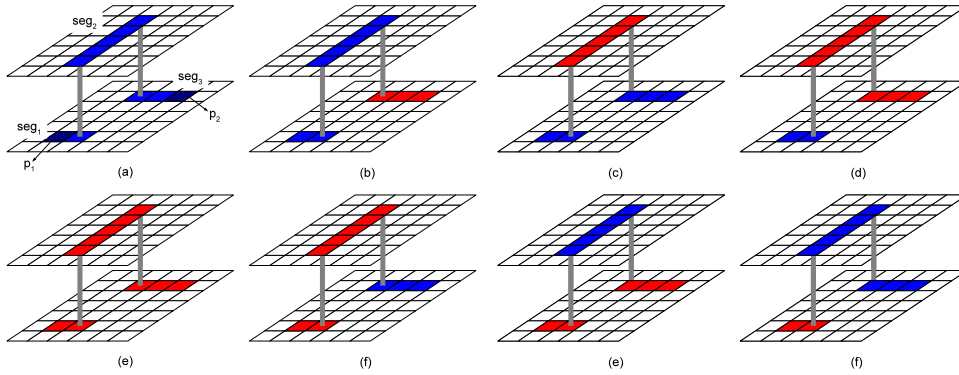


Figure 10: Solution candidates for multi-layer SADP.

simply explore all possible solutions during three-dimensional path finding. We find that, in fact, it is not necessary to maintain all combination of  $R(path_i, LD(path_i))$  during simultaneous routing and coloring. Given this observation, we develop an efficient three-dimensional path finding based on dynamic programming.

Assume a grid  $g_i$  is considered to be routed by a 2-pin net  $n(g_s, g_t)$  where  $g_s$  and  $g_t$  are the source and sink pins, respectively. We first evaluate the costs of assigning  $g_i$  as a mandrel pattern and as a trim pattern. According to the definition in Equation 1 and 2, we then obtain the accumulated cost along the path from  $g_s$  to  $g_i$ . Although there are many solution candidates for the routing path through  $g_i$ , we only need to maintain two solutions,  $cost_i(m)$  and  $cost_i(t)$ , where  $cost_i(m)$  and  $cost_i(t)$  are the accumulated routing costs when  $g_i$  is assigned as a mandrel pattern and a trim pattern, respectively. By keeping the minimum  $cost_i(m)$  and  $cost_i(t)$  in  $path_{s,i}$  for each traversed grid  $g_i$ , we are guaranteed to obtain the minimum cost solution for  $path_{s,t}$ . The solution for the routing path of  $n(g_s, g_t)$  can be expressed as the following recursive form of dynamic programming:

$$R(path_{s,t}, LD(path_{s,t})) = R(path_{s,i}, LD(path_{s,i})) + R(path_{i,t}, LD(path_{i,t})) \quad (4)$$

, for any  $g_i$  in the routing grid

According to Equation 4, we only need to maintained two minimum cost solutions  $cost_i(m)$  and  $cost_i(t)$  for any grid  $g_i$  traversed during A\* search. This makes our three-dimensional path finding more efficient on both time and space.

## 5. EXPERIMENTAL RESULTS

We implemented the proposed algorithm in C++ and tested it on the machine with 2.66GHz CPU and 4G memory. The parameters in Equation 1 and Equation 2 are set as follows:  $\alpha = \beta = 1$  and  $\gamma = 0.3$ . Two experiments test the performance and robustness of our approach. The first experiments contains only single-layer and obstacle-free designs, while the second experiment includes multi-layer designs in the presence of obstacles. For single-layer design, the method in [16] is implemented and compared with our approach. For multi-layer designs, our results are compared with a wirelength-driven routing method.

First we compare our result with [16] which simply works for single-layer designs. Because [16] also adopts A\* search technique, we are able to incorporate its cost function into our routing flow. However, due to the unavailability of the

Table 2: Benchmark statistics for multi-layer designs.

Circuit	Size( $um^2$ )	#Nets	#Blockages		
			M1	M2	Tot
CK1	20x20	29	279	26	305
CK2	48x48	306	3528	210	3699
CK3	100x100	872	13207	766	13813
CK4	160x160	1937	38792	2029	40370

benchmark in [16], we randomly generate test cases to perform the comparison. Four cases are generated with different number of nets as shown in Table 1. Note that the layout size of these cases is the same; in which Case1 has the lowest routing density while Case4 has the highest routing density. We compare the result in terms of wirelength (WL) and double patterning performance including (1) the number of spacer-protected trim patterns (#SP-trim), (2) the number of non-spacer-protected trim patterns (#NSP-trim), (3) the number of forbidden grids (#FORB grid), and (4) the number of conflicts (#conflict). The result shows our approach consistently generates better pattern quality with only a 3% wirelength increase. On average, our result generates 51% more spacer-protected trim patterns than [16], in which spacer protection implies better pattern quality. In addition, we reduce the number of non-spacer-protected trim patterns and forbidden grids by 39% and 55%, respectively.

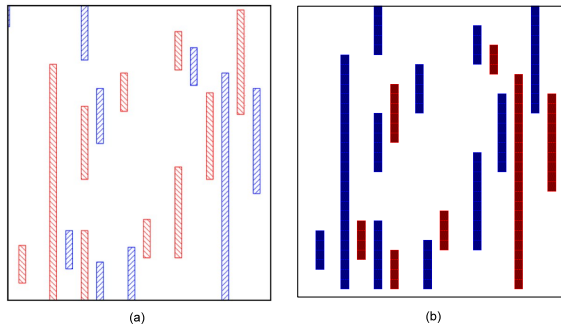
We then test the performance of our approach on multi-layer designs in the presence of blockages. Since there is no previous routing work taking double patterning into consideration on multi-layer designs, we implement a multi-layer wirelength-driven routing method followed by SADP layout decomposition as our comparison baseline. A set of two-layer industrial designs are scaled down to 22nm technology for the experiment. Table 2 gives the statistics of these designs. Each design contains two metal layers, M1 and M2, and blockages appear on both layers. Table 3 shows the comparison between our approach and the wirelength-driven routing in terms of wirelength, the number of vias (#Via), double patterning performance and runtime. Our approach achieves a great improvement in the results of double patterning. On average, the number of spacer-protected trim is increased by 2.87X; and the number of non-spacer-protected trim patterns and forbidden grids are reduced by 31% and 49%, respectively. The runtime of WL-driven is less than our approach because it does not perform any decomposability

**Table 1: Result comparison with [16] on single-layer designs.**

Testcase	#Nets	Router	WL	Double Patterning			
				#SP-trim	# NSP-trim	# FORB grid	#conflict
Case1	300	[16]	3770	28	63	3	0
		Ours	3820	40	33	0	0
Case2	600	[16]	7258	209	346	26	0
		Ours	7330	250	216	12	0
Case3	800	[16]	9704	427	727	48	0
		Ours	10130	725	464	25	0
Case4	1000	[16]	12171	750	1107	122	0
		Ours	12929	1291	702	101	0
Avg Ratio			1.03	1.51	0.61	0.45	1

**Table 3: Result comparison of routing and layout decomposition on multi-layer designs.**

Circuit	Router	WL	#Via	Double Patterning				Runtime(s)
				#SP-trim	# NSP-trim	# FORB grid	#conflict	
CK1	WL-driven	22911	48	320	262	13	22	2.3
	Ours	23045	60	480	179	3	15	6.6
CK2	WL-driven	126215	616	2397	5248	794	251	37.8
	Ours	133893	906	9397	3539	518	136	208
CK3	WL-driven	530555	1788	6222	10588	1772	757	190.8
	Ours	536215	2292	18162	7491	923	290	1021.6
CK4	WL-driven	1269046	4484	13740	25005	4375	1682	556.2
	Ours	1297775	5708	43238	17587	2787	670	2802.5
Avg Ratio		1.02	1.32	2.87	0.69	0.51	0.50	4.69


**Figure 11: Sample Layout decomposition result by (a) [9] and (b) our approach.**

checking. It is worth mentioning that the benchmarks are quite dense and some areas contain congested pins which are difficult for double patterning technology. Table 3 also shows that unresolvable conflicts exist in both of our result and wirelength-driven result, which may be fixed by post-routing techniques. Our approach outperforms wirelength-driven routing with fewer conflicts. The number of vias is increased by 32% because we utilize layer assignment to prevent conflicts and to improve the pattern quality.

Fig. 11 shows a 1D layout generated by SADP-friendly layout decomposition [9] and our approach. Our result tends to generate more mandrel patterns and reduces the number of non-spacer-protected trim patterns, which implies our result obtains better pattern quality according to the proposed routing guidelines.

Overall, our approach consistently achieves SADP-compliant results with negligible wirelength overhead. We provide more flexibility on layout decomposition by taking layer assignment into consideration. In addition, our prescribed lay-

out planning techniques greatly improve the pattern quality and thus can benefit lithography manufacturing for SADP.

## 6. CONCLUSION

In this paper, we propose a novel multi-layer SADP-aware detailed routing approach. A set of SADP-aware routing guidelines are presented, which improves SADP compliancy. We adopt a multi-layer routing model and present simultaneous layer assignment to increase the flexibility of SADP layout decomposition. Our work simultaneously solves routing and layout decomposition problems using a correct-by-construction methodology. The experimental results show that the proposed approach achieves promising results on both single-layer and multi-layer designs.

## 7. ACKNOWLEDGEMENTS

This work is supported in part by NSF, Oracle, and NSFC.

## 8. REFERENCES

- [1] Jo Finders, Mireca Dusa, and Stephen Hsu. Double patterning lithography: The bridge between low k1 ArF and EUV. *Micro lithography World*, February 2008.
- [2] George E. Bailey, Alexander Trichtkov, Jea-Woo Park, Le Hong, Vincent Wiaux, Eric Hendrickx, Staf Verhaegen, Peng Xie, and Janko Versluijs. Double pattern EDA solutions for 32nm HP and beyond. In *Proc. of SPIE*, volume 6521, 2007.
- [3] A. B Kahng, C.-H. Park, X. Xu, and Hailong Yao. Layout decomposition for double patterning lithography. In *Proc. Int. Conf. on Computer Aided Design*, November 2008.
- [4] Jae-Seok Yang, K. Lu, Minsik Cho, Kun Yuan, and D. Z. Pan. A new graph-theoretic, multi-objective layout decomposition framework for double patterning lithography. In *Proc. Asia and South Pacific Design Automation Conf.*, January 2010.
- [5] Szu-Yu Chen and Yao-Wen Chang. Native-conflict-aware wire perturbation for double patterning technology. In *Proc. Int. Conf. on Computer Aided Design*, November 2010.

- [6] Kun Yuan and D. Z. Pan. WISDOM: wire spreading enhanced decomposition of masks in double patterning lithography. In *Proc. Int. Conf. on Computer Aided Design*, November 2010.
- [7] Martin Drapeau, Vincent Wiaux, Eric Hendrickx, Staf Verhaegen, and Takahiro Machida. Double patterning design split implementation and validation for the 32nm node. In *Proc. of SPIE*, volume 6521, 2007.
- [8] Yuansheng Ma, Jason Sweis, Chris Bencher, Huixiong Dai, Yongmei Chen, Jason P. Cain, Yunfei Deng, Jongwook Kye, and Harry J. Levinson. Decomposition strategies for self-aligned double patterning. In *Proc. of SPIE*, 2010.
- [9] Yongchan Ban, K. Lucas, and D. Z. Pan. Flexible 2D layout decomposition framework for spacer-type double patterning lithography. In *Proc. Design Automation Conf.*, June 2011.
- [10] Michael C. Smayling, Christopher Bencher, Hao D. Chen, Huixiong Dai, and Michael P. Duane. APF pitch-halving for 22nm logic cells using gridded design rules. In *Proc. of SPIE*, volume 6925, 2008.
- [11] Yayi Wei and Robert L. Brainard. Advanced processes for 193-nm immersion lithography. In *Proc. of SPIE*, February 2009.
- [12] Yue Xu and C. Chu. GREMA: graph reduction based efficient mask assignment for double patterning technology. In *Proc. Int. Conf. on Computer Aided Design*, November 2009.
- [13] Hongbo Zhang, Yuelin Du, Martin D. F. Wong, and Rasit Topaloglu. Self-aligned double patterning decomposition for overlay minimization and hot spot detection. In *Proc. Design Automation Conf.*, 2011.
- [14] Minsik Cho, Yongchan Ban, and D. Z. Pan. Double patterning technology friendly detailed routing. In *Proc. Int. Conf. on Computer Aided Design*, November 2008.
- [15] Kun Yuan, K. Lu, and D. Z. Pan. Double patterning lithography friendly detailed routing with redundant via consideration. In *Proc. Design Automation Conf.*, July 2009.
- [16] Minoo Mirsaedi, J. Andres Torres, and Mohab Anis. Self-aligned double-patterning (SADP) friendly detailed routing. In *Proc. of SPIE*, 2011.
- [17] Kevin Lucas, Christopher Cork, Alexander Miloslavsky, Gerard Luk-Pat, Levi Barnes, John Hapli, John Lewellen, Greg Rollins, Vincent Wiaux, and Staf Verhaegen. Double-patterning interactions with wafer processing, optical proximity correction, and physical design flows. *Journal of Micro/Nanolithography, MEMS and MOEMS*, 8, 2009.
- [18] A. Guttma. R-Trees: a dynamic index structure for spatial searching. In *Proc. SIGMOD Int. Conf. on Management of data*, 1984.