

# Double Patterning-Aware Detailed Routing with Mask Usage Balancing

Seong-I Lei

Department of Computer Science  
National Tsing Hua University  
HsinChu, Taiwan  
Email: d9762804@oz.nthu.edu.tw

Chris Chu

Department of Electrical and Computer Engineering  
Iowa State University  
Ames, IA  
Email: cnchu@iastate.edu

Wai-Kei Mak

Department of Computer Science  
National Tsing Hua University  
HsinChu, Taiwan  
Email: wkmak@cs.nthu.edu.tw

**Abstract**—Double patterning lithography (DPL) for 32nm and 22nm technology nodes requires decomposing a layout into two masks for lithography. It is important to consider DPL during the detailed routing stage so that the layout can be decomposed easily with the minimum number of stitches. In this paper, we propose a double patterning-aware detailed routing algorithm to balance the mask usage. Different from previous works, we first fix the color of each track in the routing grid and perform detailed routing using these pre-colored tracks. Experimental results demonstrate that our algorithm yields a significant improvement on the number of stitches and decomposability.

**Keywords**—Double patterning, Detailed routing, Mask usage balancing

## I. INTRODUCTION

As the technology nodes scale down to 32nm and 22nm, single-exposure 193nm lithography has reached its printing limit and is difficult to print patterns under 32nm technology node and beyond. Double patterning lithography (DPL) [1] has emerged as a practical lithography technique for 32nm and 22nm technology nodes. In DPL, a layout is decomposed into two masks for two exposure steps as shown in Fig. 1. If the distance of two features is smaller than the minimum spacing required, these two features must be printed using different masks. Moreover, DPL can decrease the feature density in each exposure step, increase the pitch size in the mask and improve the depth of focus (DOF). For example, if we decompose a layout into two masks like Fig. 1, the pitch size in the mask can be doubled after the decomposition and therefore the printability of the layout can be improved.

The double patterning decomposition problem can be modeled as a two-color assignment problem and has been widely studied in the literature [2], [3], [4], [5], [6]. Each feature in the layout is assigned to one color and the features assigned to the same color means that they are printed using the same mask. Coloring conflict may happen during the decomposition of a layout as shown in Fig. 2(a). Three polygons *A*, *B* and *C* need to be assigned to different colors because the distance between each other is smaller than the minimum spacing required. However, only two colors are available so *A* and *C* may be assigned to the same color and therefore it leads to a coloring conflict. Using a stitch on polygon *C* that splits it into two parts of different colors may prevent such a coloring conflict as shown in Fig. 2(b). But a stitch is highly sensitive to overlay error and may lead to yield loss. It is discouraged to use too

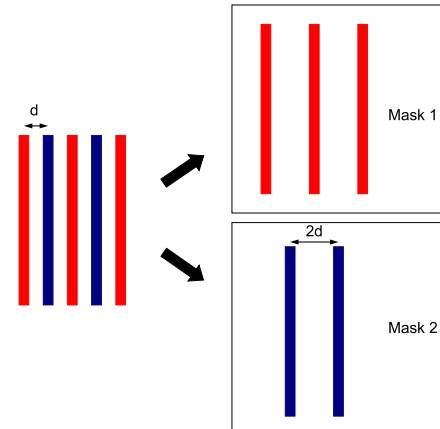


Fig. 1. Layout decomposition in DPL.

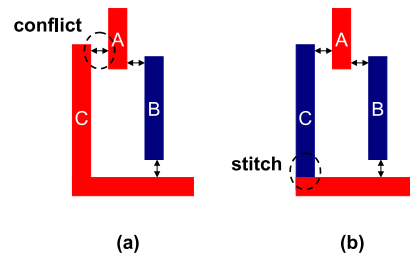


Fig. 2. (a) Coloring conflict in layout decomposition. (b) A stitch can resolve the coloring conflict.

many stitches and the number of stitches need to be minimized in the layout decomposition.

Recently, layout decomposition for triple patterning lithography (TPL) has been proposed [7], [8]. It decomposes a layout into three masks for lithography and can further increase the pitch size in each mask. It can also avoid some coloring conflicts that cannot be resolved even using stitches in DPL. However, there are some drawbacks of TPL when comparing with DPL such as an extra mask and an extra exposure step are needed in TPL. An extra mask in lithography will increase the cost of a chip because a mask is very expensive while an extra exposure step will also increase the manufacturing time of a chip. Due to these drawbacks in TPL, DPL is still preferred when it is feasible.

Considering DPL in early design phase such as detailed routing stage can increase the decomposability of the layout. Some previous works [9], [10], [11], [12] started to consider

This work was supported in part by the National Science Council under Grant NSC 102-2220-E-007-013.

DPL in detailed routing. [9] first proposes a DPL friendly detailed routing algorithm. They color the path of a net after the net is routed and shadow the surrounding grids of the colored path according to the minimum spacing required. Some grids may be uncolorable due to the dense routing paths of the nets. [10] enhances the routing algorithm proposed in [9]. They delay the coloring and shadowing until more information is available using a conflict graph. [11] enhances [9] by using an online conflict resolution algorithm to exploit the conflict graph and decrease the number of conflicts. Note that [9], [10], [11] all color the path of a net after the net is routed and may result in large number of uncolorable conflicts and stitches. [12] proposes a detailed routing scheme that considers both DPL and TPL. They propose a graph model to model the cost of conflicts and stitches in TPL and route each net by finding the shortest path of the net in the graph model. However, they need to split each edge to three edges for the three masks and the time complexity will become high if the routing grid is large.

In this paper, we propose a double patterning-aware detailed routing algorithm to balance the mask usage. Different from most previous works that color the path of a net after the net is routed. We first construct a routing grid and fix the color of each track in this grid. Then we determine the pin coloring and perform maze routing to minimize the number of stitches using the pre-colored tracks. Our contributions are summarized as follows:

- Our algorithm can take advantage of preferred routing direction assumption to guarantee to obtain a result with zero stitch if a design can be successfully routed.
- Besides minimizing the number of stitches, we also balance the mask usage such that it can enhance the printability of the layout during manufacturing. It is shown in [6] that a balanced decomposition of a layout has lower edge placement error (EPE) than an unbalanced decomposition in lithography simulation.
- Experimental results demonstrate that our algorithm yields a significant improvement on the number of stitches and decomposability compared with a recent work [12].

The rest of this paper is organized as follows. Section 2 formulates the double patterning-aware detailed routing problem. Section 3 presents our proposed algorithm. Section 4 reports the experimental results and Section 5 concludes the entire work.

## II. PROBLEM FORMULATION

We use two colors (RED and BLUE) to represent the two masks used in double patterning decomposition. We assume that each net is a two-pin net because a multi-pin net will be decomposed into two-pin nets in detailed routing. The double patterning-aware detailed routing problem is defined as follows:

### Double Patterning-Aware Detailed Routing Problem:

*Given a netlist, a routing area, a minimum spacing requirement and two colors (RED and BLUE), we need to perform detailed routing and decompose the layout into two colors such that the*

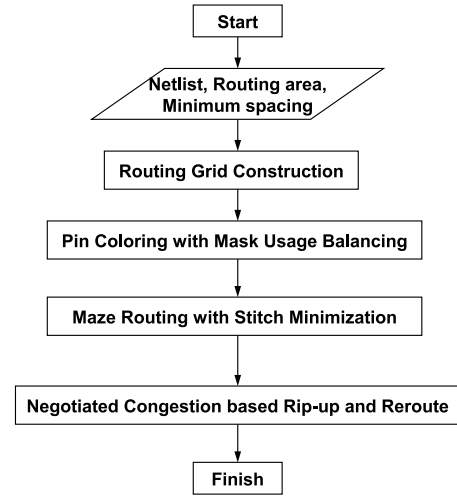


Fig. 3. The overall flow of our algorithm.

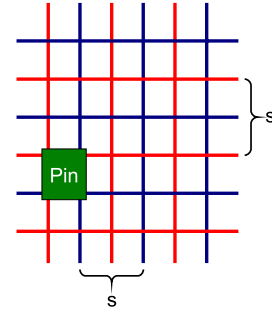


Fig. 4. The routing grid used by our algorithm. The distance  $s$  between the same color tracks must be greater than the minimum spacing required.

*total wirelength, the number of stitches, the number of vias and the number of coloring conflicts are minimized.*

## III. PROPOSED ALGORITHM

### A. Overall Flow of Our Algorithm

The overall flow of our algorithm is shown in Fig. 3. Given a netlist, a routing area and a minimum spacing requirement, we first construct the routing grid and fix the color of each track. Then we use an integer linear programming (ILP) based method to determine the pin coloring. In the pin coloring stage, we prefer the two pins of a net to use the same coloring assignment such that there may not exist any stitch for this net. We also balance the mask usage of the two colors in this stage. Next we use maze routing to route each net and only use the tracks of same color in routing according to the coloring assignment in the pin coloring stage as much as possible. Here, the number of stitches and the number of vias are minimized in the maze routing stage. Finally, we perform negotiated congestion based rip-up and reroute to improve the routing quality.

### B. Routing Grid Construction

We introduce how to construct the routing grid for detailed routing in this subsection. We construct a uniform grid for

TABLE I. PARAMETERS AND VARIABLES USED IN THE ILP FORMULATION

Parameters	
$N$	the total number of nets
$\alpha$	the lower bound of the coloring ratio
Variables	
$p_a$	0-1 integer variable that $p_a = 1$ if pin $a$ is assigned to a RED track and $p_a = 0$ otherwise
$s_j$	0-1 integer variable that $s_j = 1$ means net $j$ will have at least one stitch and $s_j = 0$ otherwise

routing and the color of each track is defined before routing. Each track is assigned either RED or BLUE color and adjacent tracks in horizontal (or vertical) direction must be assigned different colors as shown in Fig. 4. The distance  $s$  between two adjacent same coloring tracks must be greater than the minimum spacing required to avoid coloring conflict in double patterning-aware routing. For non-preferred routing direction, if the routing path of a net consists of only one color of tracks (either RED or BLUE), no stitch is required for this net. On the other hand, if the routing path of a net jumps from a RED track to a BLUE track or vice versa, a stitch is introduced for this net. However, if preferred routing direction is assumed, no stitch will occur in the routing path of a net using this routing grid. We take advantage of the fact that each pin covers a subset of grid points so that we can always access a pin by BLUE or RED track as desired.

### C. ILP-Based Pin Coloring with Mask Usage Balancing

In the pin coloring stage, we need to determine which color of a track that a pin will use in later routing stage. Here we discuss the situation under non-preferred routing direction assumption. If the two pins of a net use the different color of tracks, a stitch will have a good possibility of occurrence later in routing. Therefore, we prefer the two pins of a net to use the same color of tracks such that there may not have any stitch in routing. We also want to balance the mask usage of the two colors in this stage. We expect that the number of the RED tracks used is similar to the number of the BLUE tracks used. We use an ILP-based method to determine the pin coloring. The parameters and variables used in our ILP formulation are defined in Table I.

Our ILP formulation of pin coloring is as follows:

$$\min \sum_{j=1}^N s_j$$

$$s.t. \quad p_a - p_b \leq s_j \quad \forall \text{net } j(a, b) \quad (1)$$

$$p_a - p_b \geq -s_j \quad \forall \text{net } j(a, b) \quad (2)$$

$$\alpha \leq \sum_{j=1}^N \frac{0.5HPWL_j}{totalHPWL} (p_a + p_b) \leq 1 - \alpha \quad (3)$$

$$p_i, s_j \in \{0, 1\} \quad \forall i, j \quad (4)$$

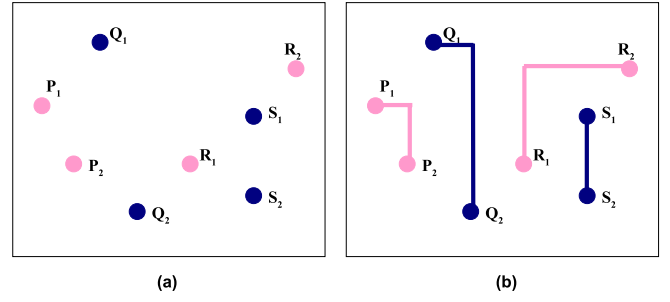


Fig. 5. (a) A pin coloring assignment after solving the ILP. (b) The potential routing result according to the pin coloring assignment shown in (a).

As we cannot get the information of layer assignment in this stage, we assume single layer is used in this ILP formulation. We assume pin  $a$  and pin  $b$  belong to net  $j$ , constraints (1) and (2) mean that if the color assignment of pin  $a$  and that of pin  $b$  are the same, there will be no stitch for this net. Otherwise, there will be at least one stitch for this net. If  $p_a = 1$  and  $p_b = 0$ , constraint (1) will force  $s_j$  to be one. If  $p_a = 0$  and  $p_b = 1$ , constraint (2) will force  $s_j$  to be one. On the other hand, if  $p_a = p_b = 0$  or  $p_a = p_b = 1$ ,  $s_j$  will be zero because we minimize  $s_j$  in the objective function.

We balance the mask usage using constraint (3). We use half-perimeter wirelength (HPWL) to estimate the wirelength of each net.  $HPWL_j$  means the HPWL of net  $j$  and  $totalHPWL$  is the total HPWL of all nets. We define the coloring ratio to be the ratio of the total HPWL in RED mask to the total HPWL in the two masks. If the two pins of a net are both assigned to RED mask, it will add the HPWL of this net to the total HPWL in RED mask. If only one pin of a net is assigned to RED mask, it means there is at least one stitch for this net and we assume the stitch is located near the middle of this net, so on average half of the HPWL of this net is incurred on the RED mask. The coloring ratio is bounded by a lower bound  $\alpha$  and an upper bound  $1 - \alpha$  for balancing the two masks in constraint (3).  $p_i$  and  $s_j$  are the 0-1 integer variables in our ILP formulation. Finally, the total number of stitches is minimized in the objective function.

We use the example in Fig. 5 to show how the pin coloring stage can help to reduce the number of stitches and balance the mask usage. In Fig. 5(a), after solving the ILP, the pins of nets  $P$  and  $R$  are assigned to the RED mask and the pins of nets  $Q$  and  $S$  are assigned to the BLUE mask. Fig. 5(b) shows a potential routing result according to this pin coloring assignment. As the number of stitches is minimized in the ILP formulation, the two pins of a net are assigned to the same color and we will route the net using only the tracks with this color. Nets  $P$  and  $R$  are routed in the RED mask and nets  $Q$  and  $S$  are routed in the BLUE mask. The mask usage of the routing result is also well balanced.

### D. Maze Routing with Stitch Minimization

We use A\* search algorithm to perform the maze routing for minimizing the number of stitches and vias in this stage. Our maze routing algorithm is shown in Algorithm 1.

We use a heap  $H$  to hold the solutions and propagate the solutions from source to sink (lines 6-26). Each time a

---

**Algorithm 1** Maze Routing with Stitch Minimization

---

**Require:** A routing area, a netlist

```
1: for each net  $i$  in netlist do
2:    $s \leftarrow$  source grid point of  $i$ 
3:    $t \leftarrow$  sink grid point of  $i$ 
4:    $s.cost \leftarrow 0$ 
5:   insert  $s$  to a heap  $H$ 
6:   while  $H$  is not empty do
7:      $current \leftarrow$  extract-min( $H$ )
8:     if  $current = t$  then
9:       terminate and return the solution
10:    end if
11:    for each neighboring grid point  $n$  of  $current$  do
12:       $cost \leftarrow current.cost + 1 + MD(n, t)$ 
13:      if  $current$  and  $n$  are in the same layer then
14:        if  $current.track\_color \neq n.track\_color$  then
15:           $cost \leftarrow cost + Stitch\_cost$ 
16:        end if
17:      end if
18:      if  $current$  and  $n$  are not in the same layer then
19:         $cost \leftarrow cost + Via\_cost$ 
20:      end if
21:      if  $cost < n.cost$  then
22:         $n.cost \leftarrow cost$ 
23:        insert  $n$  to  $H$ 
24:      end if
25:    end for
26:  end while
27: end for
```

---

minimum cost solution is extracted from  $H$  (line 7). If the sink is reached, the algorithm will terminate and return the solution (lines 8-9). Otherwise, we propagate the solution to the neighboring grid points of the current solution (lines 11-25). In line 12, we use the Manhattan distance from the neighboring grid point to the sink ( $MD(n, t)$  in Algorithm 1) as our A\* heuristic cost and each step of propagation will increase one unit wirelength in the cost function. We increase the cost function with a stitch cost if the propagation from the current grid point to the neighboring grid point has a different track color (lines 14-16). The cost function is increased with a via cost if the solution is propagated from one layer to another layer (lines 18-20). Finally, we update the cost of the neighboring grid point and insert the solution of the neighboring grid point to  $H$  (lines 21-24).

In our implementation, if the two pins of a net are assigned to the same color after solving the ILP, we will perform the maze routing using only the tracks of that color. If the net can be successfully routed, there will be no stitch for this net. Otherwise, we perform the maze routing using all the tracks.

#### E. Negotiated Congestion based Rip-up and Reroute

We adopt a negotiated congestion based routing scheme [13] for improving the routing quality. Initially, all the nets are routed independently. After the initial routing, some routing resources may be shared by multiple nets, and we use the negotiated congestion based routing scheme to eliminate these illegal sharings. The history cost of the shared resources will be increased by one in each iteration and the nets in congested

regions will tend to spread out and look for other routing paths in the regions with less or no congestion. During each iteration, we rip-up and reroute the nets in the congested regions until there is no illegal sharing.

## IV. EXPERIMENTAL RESULTS

We implemented our algorithm in the C++ programming language and all experiments were performed on a 3GHz Linux machine with 64GB memory. The Gurobi optimizer [14] is employed as our ILP solver. We use  $Stitch\_cost = 30$ ,  $Via\_cost = 10$  and  $\alpha = 0.49$  in our experiment. We have obtained the source code of [12] for comparison. Testcases C1 and C2 are provided by [12] and we further generate other six testcases. For testcases C1-C4, each pin covers only one grid point. For testcases T1-T4, each pin covers multiple grid points (2-5 grid points). We compare our algorithm with [12] and the results of comparison under non-preferred direction assumption and under preferred direction assumption are shown in Table II and Table III, respectively. Wirelength (WL), the ratio of WL in RED mask to the total WL (RED ratio), number of stitches, number of coloring conflicts, number of vias and runtime are reported.

For the experiment under non-preferred routing direction assumption shown in Table II, two routing layers are assumed for each testcase and each layer has both horizontal and vertical routing tracks. Our approach results in 1.9% shorter wirelength, 85.2% less stitches, 98.2% less coloring conflicts and 18.8% more vias. As [12] needs to split each edge of their routing graph into two edges for the two masks and they need to perform rip-up and reroute to reduce coloring conflicts, the runtime of their algorithm is very large. Our algorithm is about 6 times faster than [12]. The runtime of our ILP formulation of pin coloring in all test cases is less than two seconds.

For the experiment under preferred routing direction assumption shown in Table III, four routing layers are assumed for each testcase and each layer has either horizontal or vertical routing tracks. Our algorithm can achieve 0.2% shorter wirelength and 9% less vias. Most importantly, our algorithm does not have any stitch and coloring conflict while [12] still has a certain number of stitches. This is because [12] allows a routing path to change colors in a horizontal (or vertical) track. However, our algorithm only allows a routing path to change colors from a horizontal track to a vertical track (or vice versa) and these two types of tracks are on different layers under preferred routing direction assumption. Therefore, no stitch will occur in our algorithm. In other words, we can always obtain a result with zero stitch and zero coloring conflict under preferred routing direction assumption. Moreover, the large size of the routing graph in [12] will affect the performance greatly when more layers are used. Our algorithm is about 43 times faster than [12].

We note that [12] balances the usage of BLUE mask and RED mask by dynamically adjusting the cost of BLUE edges according to the current ratio of the wirelength in BLUE over the wirelength in RED as routing proceeds. On the other hand, our approach allows a flexible targeted range of RED color ratio. In the experiments, the targeted range was set to 0.49-0.51 and the experimental results show that constraint (3) is already an effective means to control the final RED color ratio to the desired range.

TABLE II. COMPARISON OF THE RESULTS UNDER NON-PREFERRED DIRECTION ASSUMPTION BETWEEN [12] AND OUR ALGORITHM

Testcases	#Net	Grid size	[12]						Ours					
			WL	RED ratio	#Stitch	#Conflict	#Via	Runtime (s)	WL	RED ratio	#Stitch	#Conflict	#Via	Runtime (s)
C1	1500	100×100	10840	0.500	1115	3874	2322	251	10110	0.500	108	31	2124	88
C2	10000	300×300	69755	0.499	1025	660	8492	18069	62193	0.505	156	27	11544	30
C3	1927	400×400	67087	0.505	113	0	1518	9710	65711	0.485	70	13	1950	2660
C4	2400	400×400	67777	0.503	157	0	1626	4921	66307	0.503	71	10	2198	1609
T1	869	500×500	100190	0.497	57	0	1360	93019	99306	0.506	0	0	1244	48802
T2	1036	600×600	129719	0.518	70	0	1550	278614	128611	0.512	2	0	1510	76161
T3	1763	800×800	216739	0.499	124	0	2576	752751	215256	0.499	0	0	2460	99911
T4	3017	1000×1000	197244	0.496	83	0	2136	390017	195534	0.529	0	0	2610	47744
average			107418.9	0.502	343.0	566.8	2697.5	193419.0	105378.5	0.505	50.9	10.1	3205.0	34625.6
ratio			1.019		6.739	56.119	0.842	5.586	1		1	1	1	1

TABLE III. COMPARISON OF THE RESULTS UNDER PREFERRED DIRECTION ASSUMPTION BETWEEN [12] AND OUR ALGORITHM

Testcases	#Net	Grid size	[12]						Ours					
			WL	RED ratio	#Stitch	#Conflict	#Via	Runtime (s)	WL	RED ratio	#Stitch	#Conflict	#Via	Runtime (s)
C1	1500	100×100	9636	0.497	59	5	5274	773	9054	0.491	0	0	3900	11
C2	10000	300×300	62935	0.501	278	3	27370	33587	60325	0.498	0	0	23036	60
C3	1927	400×400	64543	0.499	25	0	4180	883	64347	0.500	0	0	4084	42
C4	2400	400×400	64535	0.489	34	0	5236	1164	64331	0.498	0	0	5124	42
T1	869	500×500	98898	0.505	10	0	1962	1810	99146	0.509	0	0	2092	122
T2	1036	600×600	128121	0.508	8	0	2264	7511	128429	0.504	0	0	2480	819
T3	1763	800×800	214467	0.495	7	0	3754	20069	215035	0.504	0	0	4060	539
T4	3017	1000×1000	193768	0.495	6	0	6072	10538	194716	0.513	0	0	6306	159
average			104612.9	0.499	53.4	1.0	7014.0	9541.9	104422.9	0.502	0	0	6385.3	224.3
ratio			1.002				1.098	42.550	1				1	1

## V. CONCLUSION

In this paper, we propose a double patterning-aware detailed routing algorithm to balance the mask usage. Different from previous works, we first fix the color of each track in the routing grid and perform detailed routing using these pre-colored tracks. The mask usage is balanced in the proposed ILP-based pin coloring stage and a flexible targeted range is allowed to control the color ratio. Experimental results show that our efficient algorithm yields a significant improvement on the number of stitches and decomposability. For future work, we may consider TPL or multiple patterning lithography in our detailed routing algorithm.

## REFERENCES

- [1] G. E. Bailey, A. Tritchkov, J. W. Park, L. Hong, V. Wiaux, E. Hendrickx, S. Verhaegen, P. Xie and J. Versluis, "Double pattern EDA solutions for 32nm HP and beyond," in *Proc. SPIE*, vol. 6521, 2007.
- [2] A. B. Kahng, C. H. Park, X. Xu and H. Yao, "Layout decomposition for double patterning lithography," in *Proc. ICCAD*, pp. 465-472, 2008.
- [3] K. Yuan, J. S. Yang and D. Z. Pan, "Double patterning layout decomposition for simultaneous conflict and stitch minimization," in *Proc. ISPD*, pp. 107-114, 2009.
- [4] Y. Xu and C. Chu, "GREMA: Graph reduction based efficient mask assignment for double patterning technology," in *Proc. ICCAD*, pp. 601-606, 2009.
- [5] X. Tang and M. Cho, "Optimal layout decomposition for double patterning technology," in *Proc. ICCAD*, pp. 9-13, 2011.
- [6] J. S. Yang, K. Lu, M. Cho, K. Yuan and D. Z. Pan, "A new graph-theoretic, multi-objective layout decomposition framework for double patterning lithography," in *Proc. ASPDAC*, pp. 637-644, 2010.
- [7] B. Yu, K. Yuan, B. Zhang, D. Ding and D. Z. Pan, "Layout decomposition for triple patterning lithography," in *Proc. ICCAD*, pp. 1-8, 2011.
- [8] S. Y. Fang, Y. W. Chang and W. Y. Chen, "A novel layout decomposition algorithm for triple patterning lithography," in *Proc. DAC*, pp. 1181-1186, 2012.
- [9] M. Cho, Y. Ban and D. Z. Pan, "Double patterning technology friendly detailed routing," in *Proc. ICCAD*, pp. 506-511, 2008.
- [10] X. Gao and L. Macchiarulo, "Enhancing double-patterning detailed routing with lazy coloring and within-path conflict avoidance," in *Proc. DATE*, pp. 1279-1284, 2010.
- [11] I. Abed and A. G. Wassal, "Double-patterning friendly grid-based detailed routing with online conflict resolution," in *Proc. DATE*, pp. 1475-1478, 2012.
- [12] Q. Ma, H. Zhang and M. D. F. Wong, "Triple patterning aware routing and its comparison with double patterning aware routing in 14nm technology," in *Proc. DAC*, pp. 591-596, 2012.
- [13] L. McMurchie and C. Ebeling, "Pathfinder: a negotiation-based performance-driven router for FPGAs," in *Proc. FPGA*, pp. 111-117, 1995.
- [14] Gurobi optimizer: <http://www.gurobi.com>