Detailed Routing for Spacer-Is-Metal Type Self-Aligned Double/Quadruple Patterning Lithography

Yixiao Ding
Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50010, USA
yxding@iastate.edu

Chris Chu
Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50010, USA
cnchu@iastate.edu

Wai-Kei Mak
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 30013
wkmak@cs.nthu.edu.tw

ABSTRACT

As the technology nodes scale down to 22nm and beyond, Double Patterning Lithography (DPL) has been considered as a practical solution for manufacturing process. Compared with Litho-Etch-Litho-Etch (LELE), Self-Aligned Double Patterning (SADP) has better overlay tolerance. Two types of SADP process are popularly used for the state-of-the-art lithography patterning: Spacer-Is-Dielectric (SID) and Spacer-Is-Metal (SIM). Meanwhile, Self-Aligned Quadruple Patterning (SAQP), as a natural extension of SADP, is expected to be one of the major solutions for future process requirement after the 16nm/14nm technology node. In order to have better decomposability of layout patterns, we consider SIM type SADP/SAQP during detailed routing stage. The idea of color pre-assignment is adopted and a graph model is proposed which greatly simplifies the problem and reduces design rule violation. Then, the negotiated congestion based scheme is applied for detailed routing based on our proposed graph model. Compared with other state-of-art works, our approach does not produce any side overlay error and no design rule violation is reported. Meanwhile, a better solution in terms of total wirelength, via count, routability, and runtime is achieved.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids

General Terms

Algorithms, Design, Performance

Keywords

SADP/SAQP, Routing, Manufacturability

1. INTRODUCTION

As the technology nodes scale down to 22nm and beyond, Double Patterning Lithography (DPL) has been considered as a practical solution for manufacturing process. There are two major types of DPL: Litho-Etch-Litho-Etch (LELE) and Self-Aligned Double Patterning (SADP). Given a target layout, LELE decomposes layout patterns and assigns them into two masks for two exposure steps. With an additional mask, the pattern density in each mask decreases and therefore the design rule violation can be reduced effectively. Fig. 1(b) shows an example of LELE decomposition and patterns assigned to different masks have different shading. For the SADP type, a film layer called the spacer is deposited on the sidewall of pre-featured patterns called the mandrel pattern. Since two sidewall spacers are deposited for each mandrel pattern, the pattern density now can be doubled and pitch size can be halved. Two types of SADP process are popularly used for the state-of-the-art lithography patterning: Spacer-Is-Dielectric (SID) and Spacer-Is-Metal (SIM).

DAC '15 June 07 - 11, 2015, San Francisco, CA, USA

http://dx.doi.org/10.1145/2744769.2744821

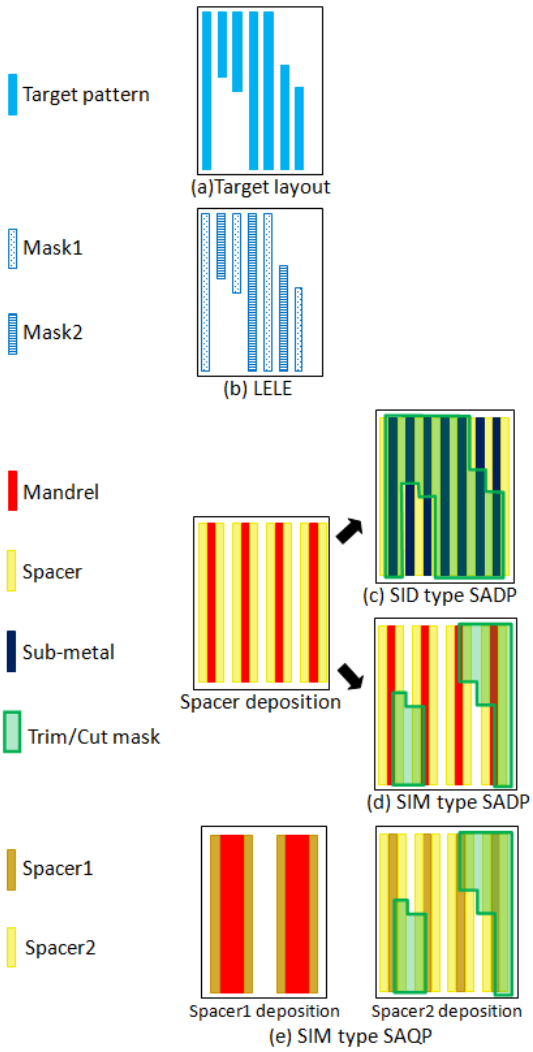

Figure 1: Overview of DPL and SAQP lithography

Compared with LELE, SADP has a great advantage in overlay tolerance. For both types of SADP, mandrels patterns are firstly formed and spacers are deposited along the sidewall of mandrel patterns. For the SID type, after spacer deposition, the mandrels are removed and sub-metals are deposited at both original mandrels locations and space between spacers. Then, trim mask is applied to help trimming out the target layout

patterns. The regions not covered by spacer but covered by trim mask produce the final patterns. For the SIM type, the sidewall spacer is made of metal and will form the final metal pattern. We use the cut mask to cut off unwanted portion of patterns formed by the spacers. Thus, the regions covered by spacer but not covered by cut mask produce the final patterns. Fig. 1(c)(d) show an example of manufacturing process for each type of SADP.

For continuous shrinking of technology nodes, Self-Aligned Triple Patterning (SATP) and Self-Aligned Quadruple Patterning (SAQP) have been proposed recently. Compared with SATP, SAQP technique turns out to be a better solution by enabling higher device density (or smaller pitch) and more robust process control [1]. It is one of major techniques for the future process requirement after 16/14nm technology node [2]. As a natural extension of SADP, the manufacturing process of SAQP is similar to SADP. The only difference is that two steps of spacer deposition are performed. Note that there are also two types of SAQP which depends on whether the spacer in the second step of deposition is dielectric or metal. Below we only describe the SIM type SAQP. Firstly, mandrel patterns are formed and spacer1 is deposited along the sidewall of mandrel patterns. Then, the mandrels are removed and spacer2 is deposited along the sidewall of spacer1. Next, spacer1 is etched away and spacer2 forms the final metal patterns. We also use the cut mask to cut off unwanted portions in order to generate the target layout. In summary, the regions covered by spacer2 but not covered by cut mask produce the final patterns. Fig. 1(e) summarizes the whole process.

A layout configuration without considering DPL at design stage can make the layout hard to decompose. Thus, the intended layout is not manufacturable and redesign requires high cost. Thus, it is necessary to consider whether the layout is DPL decomposable during design time, especially the detailed routing stage. This will significantly reduce the number of design rule violations and improve decomposability of layout patterns during manufacturing time.

There are several major works considering DPL during the detailed routing stage. [3] considered LELE type DPL and applied the idea of color pre-assignment to routing tracks. The proposed algorithm helps in reducing design rule violation significantly during detailed routing stage. [4] [5] [6] considered SID type SADP during detailed routing stage. [4] split each node of the routing grid into 4 vertices and construct an expanded routing graph model on them. This greatly slowed down the runtime and increased memory load of the proposed algorithm. In addition, they did not mention how to deal with via in their graph model. [5] enumerated all the potential overlay scenarios and used a constraint graph to detect them. The cost of this enumeration and detection is relatively high to the detailed router. Furthermore, the cut process is used which leads to side overlay error. [6] solved routing and layout decomposition problems simultaneously based on the correct-by-construction approach. However, the routing scheme heavily depends on the net ordering. Furthermore, the proposed SADP-aware routing guidelines in [6] are not fully justified as pointed out by [4]. For the SAQP, [2] presented an early study on the definition of SAQP-friendly layout and introduced some guidelines for SAQP decomposition. However, they did not consider SAQP during the detailed routing stage. [7] proposed the first grid routing method for SID type SADP/SAQP. The routing is performed on a grid structure where grid nodes are alternately colored by two colors or uncolored for SADP (by three colors or uncolored for SAQP). However, their approach is unrealistic because it requires that every pin of each net must fall on the same colored grid nodes, otherwise it cannot route the net. To the best of our knowledge, there is no previous work considering SIM type SADP/SAQP during detailed routing stage. Thus, we study this problem in the paper.

The contributions of our paper is summarized as follows:

- To the best of our knowledge, this is the first time to systematically consider SIM type SADP/SAQP lithography during detailed routing stage.
- We apply the idea of color pre-assignment to routing panels and propose a new graph model. These simplify the overall routing process and significantly reduce design rule violation.
- The proposed detailed router for SIM type SADP can be easily extended for SAQP with little modification.
- The experimental results demonstrates that SIM type SADP /SAQP is a promising option to manufacture layout.

The rest of the paper is organized as follows. Section 2 provides some preliminary information, especially the design rules of SIM type SADP/SAQP lithography. Section 3 gives our problem formulation. Section 4 presents the details of our proposed solution to the problem. Section 5 demonstrates our experimental results. Finally, Section 6 concludes the paper and Section 7 is acknowledgements.

2. PRELIMINARIES

In this section, we will discuss several manufacturing challenges when we consider SIM type SADP/SAQP lithography during detailed routing stage.

2.1 Mandrel and cut mask design rules

Two masks are used for manufacturing in SIM type SADP/ SAQP. The one used to form mandrel patterns is the mandrel mask. The other one used for cutting is the cut mask. Due to the optical resolution limits, there are several design rules imposed on these two masks. For example, minimum spacing constraints and minimum width constraints should be maintained for all the patterns on the mask. In this paper, we define S_m and W_m as the minimum spacing and minimum width values for mandrel mask. Meanwhile, we define S_c and W_c as the minimum spacing and minimum width values for cut mask. Fig. 2 (a)(b) show the design rules for both mandrel and cut mask patterns.

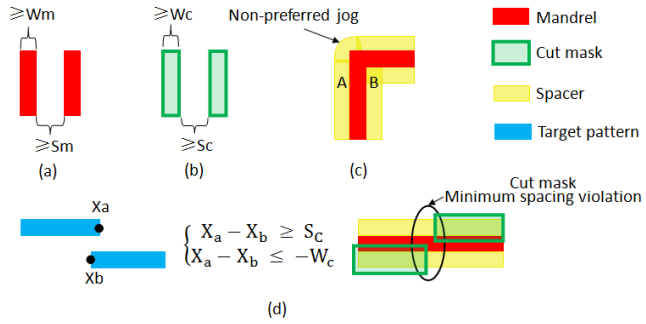

Figure 2: SIM type SADP/SAQP lithography design rules. (a)Mandrel design rules. (b)Cut mask design rules. (c)Non-preferred jog degradation. (d)Prohibited Anti-Parallel Line-Ends

2.2 Non-preferred jog

[8] states that spacer deposited at a convex mandrel corner gets rounded even with refined OPC for the mandrel mask. Instead, the shape of spacers stays sharp at concave mandrel corners. For our case of SIM type, the spacers are used to form the final metal patterns. Thus, the rounding issue of spacers will cause yield loss on the metal wire. Therefore, whenever L shape of wire patterns are formed, we prefer using spacers deposited around concave corners of mandrels. We refer it as minimization of non-preferred jog in this paper. [4] also identified this manufacturing challenge and called it sm-jogs minimization in the SID type SADP lithography. Fig. 2(c) shows an example of spacer A are deposited with a rounding issue. If the spacer A is used to form the final metal pattern, this non-preferred turn will cause yield loss.

2.3 Prohibited Anti-Parallel Line-Ends

[4] identified "Prohibited Anti-Parallel Line-Ends" as one of intrinsic challenges in SID-compliant detailed routing, which also applies to SIM type. In our case, it refers to the cut mask design rule violation caused by two anti-parallel line-ends in the target layout. As shown in Fig. 2(d), suppose the X_a and X_b are horizontal coordinates for two anti-parallel line-ends. In order to avoid cut mask design rule violation when generating target layout, either $X_a - X_b \geq S_c$ or $X_a - X_b \leq -W_c$ should be satisfied. Otherwise, minimum spacing or minimum width constraint on the cut mask will be violated.

3. PROBLEM FORMULATION

We assume that there is a preferred routing direction for each layer and the other direction perpendicular to the preferred routing direction is defined as non-preferred routing direction of the layer. We do not completely disable routing in the non-preferred routing direction. We refer to the above problem as the restricted detailed routing problem. Meanwhile, we refer the violation of manderel and cut mask design rule in Section 2.1 and prohibited anti-parallel line-ends in Section 2.3 as design rule violation. With such assumption and design rules mentioned in Section 2, we formulate the SIM type SADP/SAQP lithography compliant detailed routing problem.

Given a netlist with candidate source/sink pin locations for every net, a routing grid, and design rules, detailed routing with simultaneous pin location determination for all the net are performed. The final routing patterns should be compliant to SIM type SADP/SAQP lithography. The objective is to minimize the design rule violations, the number of unroutable nets, total wirelength, the number of vias, and non-preferred jog.

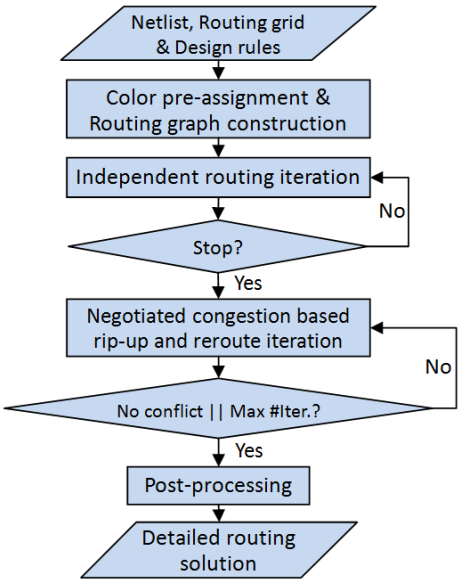

Figure 3: The overall flow of proposed algorithm

4. PROPOSED SOLUTION

4.1 Overall flow

The overall flow of our proposed algorithm is shown in Fig. 3. Assuming a netlist, a routing grid, and design rules are given. The routing of each net should be along the grid lines. We firstly perform color pre-assignment, then we build a routing graph based on our proposed graph model. After that, we perform independent routing iterations. During this phase, we route all the net *almost* independently in order to minimize the effect of routing order. Several heuristics are applied here in order to obtain better solution in fewer number of iterations. This phase will terminate if the pre-set stopping criterion is satisfied. In our implementation, we will stop if the routing congestion of current iteration is no better than the previous one. Next, we treat the output of independent routing iteration as initial routing solution for the negotiated congestion based rip-up and reroute phase. The rip-up and reroute iteration will continue until there is no conflict in the routing solution or we reach the pre-set maximum number of iterations. The last phase is post-processing in which we eliminate all the prohibited anti-parallel line-ends. Finally, we generate our detailed routing solution which is compliant to SIM type SADP/SAQP lithography.

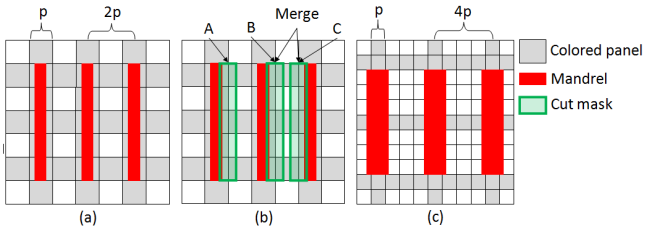

Figure 4: Color pre-assignment to routing panel. (a)(b)SADP color pre-assignment. (c) SAQP color pre-assignment.

4.2 Color pre-assignment

The SADP layout decomposition is not intuitive in the sense that the resulting mandrel mask and cut mask look significantly different from the original layout. In addition, design rules need to be considered for both mandrel and cut masks. In order to simplify the problem, we propose an idea of color pre-assignment onto the routing panels. On each layer, we define a panel as the area between two adjacent horizontal (vertical) grid lines. We pre-assign color to the panels alternately in both the horizontal and vertical direction. Fig. 4(a) shows the colored panels prepared for SADP lithography detailed routing. The colored panels specify where the mandrel patterns may be formed. Meanwhile, mandrel pattern is required to be aligned in the middle of colored panel which is shown in Fig. 4(a). Assume the pitch size of routing tracks p is given. In order to align the metal patterns along the routing tracks in SIM type SADP lithography, we require $w_m + w_s = p$ where w_m is the width of mandrel, and w_s is the width of spacer. If we require that the pitch size of colored panels, which is $2p$, is larger than or equal to $S_m + w_m$, the minimum spacing constraint on the mandrel mask are easy to be maintained. Since 193nm (ArF) wavelength light source is used for both mandrel and cut mask lithography, we assume $S_m \approx S_c$. Thus, the minimum spacing constraint on the cut mask is also automatically maintained if two cut mask patterns are aligned on the same side of mandrels. For two cut mask patterns within minimum spacing, we have the option to merge them in order to resolve the design rule violation. Fig. 4(b) shows three possible position of cut mask patterns: A, B, and C. A and B are separated with enough distance due to the requirement $2p \geq S_m + w_m$. B and C are too close. However, they can be merged without any design rule violation. SAQP has similar color pre-assignment process. As shown in Fig. 4(c), the only difference is that we assign color to every four panels both in horizontal and vertical direction. In addition, we require $4p \geq S_m + w_m$ in order to avoid design rule violation.

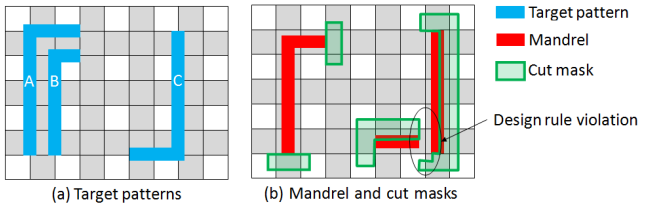

Figure 5: An example of SIM type SADP lithography

With such a restriction that all mandrel patterns are formed within colored panels, we observe that some patterns are not

manufacturable by SIM type SADP process due to design rule violation. In Fig. 5, given three "L" shape target patterns A, B, and C in (a), the corresponding mandrel and cut mask design is shown in (b). A design rule violation is detected when trying to generate pattern C. Both mandrel and cut mask patterns violate the minimum spacing constraint. Thus, pattern C is not manufacturable and is therefore forbidden during our detailed routing stage. Meanwhile, pattern A is manufactured with non-preferred jog degradation while pattern B is manufactured without such degradation. From the example above, it is observed that how a routing pattern turns at the intersection point of routing grid determines the manufacturing cost. We define the turning in pattern B with no degradation as a preferred turn, the turning in pattern A with non-preferred jog degradation as a feasible turn, and the turning in pattern C with unresolvable design rule violation as a forbidden turn. In the next subsection, we will introduce our graph model which captures all these manufacturing cost exactly.

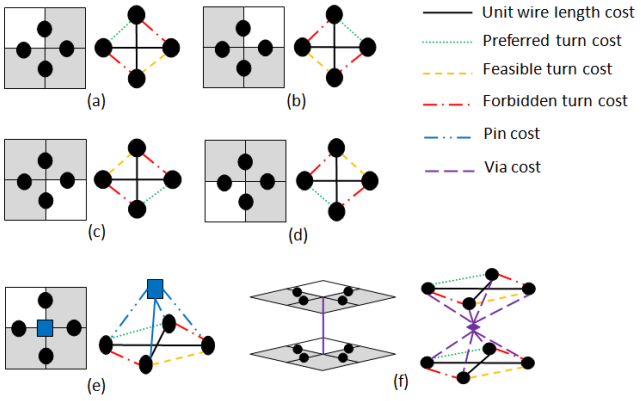

Figure 6: Graph model. (a)-(d) grid segment model. (e) pin model. (f) via model

4.3 Graph model

In this section, we introduce our graph model which helps capturing the cost of routing and manufacturing during the detailed routing stage.

Suppose we are given a multi-layer routing grid G with color pre-assignment and a preferred routing direction for each layer. We construct our routing graph G' by viewing each grid segment and via as a vertex. In addition, each pin is also viewed as a vertex in the graph. An edge exists between two vertices if they are directly connected in the routing grid. A cost is associated with each edge to indicate the cost of travelling from the vertex in one end to the vertex in the other end. The construction of G' is described below by considering graph models for grid segment, pin, and via, separately. Fig. 6 shows the three types of graph models and the six types of edges including unit wire length, preferred turn, feasible turn, forbidden turn, pin, and via. Furthermore, we consider grid segment model for four different scenarios and each scenario is identified by the relative position of the uncolored grid square. Fig. 6(a)-(d) show four different scenarios and the different types of edges in each scenario are marked by different colors. Since we consider the restricted routing problem, we simply add an additional cost to the edges which connect grid segments in the non-preferred routing direction as a penalty of non-preferred direction routing. The graph models for pin and via are more intuitive. Fig. 6(e) and (f) shows the models and corresponding edge types. SAQP has similar routing graph construction. The only difference is that 16 different scenarios should be considered for the grid segment model. The increased number of scenarios is due to the different way of color pre-assignment for SAQP. We are not expanding on the details here.

Since the proposed graph model is an undirected graph, the

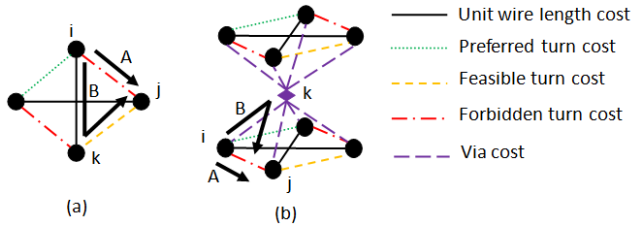

Figure 7: Graph model correction during implementation

path computed by shortest path algorithm may not be a valid route in the routing grid. There are two cases of invalid paths, which are illustrated in Fig. 7. As shown in Fig. 7(a), a path is supposed to travel from vertex i to j through path segment A, it might choose path segment B instead. The reason is that the cost of a unit wirelength edge together with a feasible edge is less than that of a forbidden edge. However, the path segment B is not a valid route in the routing grid. Similar issue also occurs in Fig. 7(b). So, we correct such issues by modifying Dijkstra's algorithm. Whenever we perform relaxation on an edge (k, j) to update the minimum distance for vertex j, we check whether vertex k's parent is adjacent to vertex j. If so, we abandon this relaxation. In this way, the path computed by the modified Dijkstra's algorithm is always a valid route in the routing grid.

4.4 Overall routing scheme

As shown in Algorithm. 1, the overall routing algorithm has three phases. We apply negotiated congestion based routing scheme in our algorithm. At the end of each iteration, we increase the costs for all the congested routing resources. In this way, the congested routing resources grow more expensive over iterations. The net with more alternative routes tends to detour from the congested area, thus the routing conflicts can be potentially resolved. Here we introduce the other two cost components used in our routing scheme.

$$Cost_e = BC_e + UC_e + HC_e^i$$

$$UC_e = \alpha \times U(S_a) + \alpha \times U(S_b) + \beta \times U(P_p)$$

$$HC_e^i = HC_e^{i-1} + \gamma \times OF(S_a) + \gamma \times OF(S_b) + \theta \times OF(P_p)$$

where $Cost_e$ is the total cost of an edge e, BC_e is the base cost in the original graph model introduced in section IV.C, UC_e is the usage cost indicating current usage, HC_e^i is the history cost after iteration i indicating historical congestion information. Furthermore, the UC_e is computed by identifying current usage of segment a, segment b, and grid point p, where segments a and b correspond to the two vertices of edge e, p is the shared grid point of a and b. Finally, HC_e^i is computed by accumulating HC_e^{i-1} with weighted overflow of segment a, segment b, and grid point p in iteration i.

The first phase is independent routing iteration. Note that since a couple of heuristics are applied here, the routing of a net is not totally independent from the previously routed nets. The heuristics are introduced below.

- Since in the final routing solution, the route of every net should not go through pin location of all other nets. Therefore, every time we route a net, we block all the routing resource which are occupied by the pins of all other nets. In this way, the potential routing conflicts will probably be reduced.

- As mentioned before, every net is routed by performing Dijkstra's algorithm from source to sink pin. It is possible that multiple optimal solutions exist for a net. Thus, choosing one optimal solution which has fewest number of conflicts with previously already routed nets will probably reduce potential routing conflicts. Therefore, every time after we route a net, we will update usage cost due to the

```
In:Netlist, routing graph which models routing resource and design rules;
Out:Routing solution and pin location determination;
Phase 1: Independent routing iteration;
while Stop criteria not satisfied do
    for each net i in the netlist do
        Block routing resource used by pins of all other nets;
        Dijkstra's algorithm to route net i;
        Update UC for net i, α and β are extremely small;
    end
    Update HC for all the congested routing resource;
    Remove UC;
end
Remove HC;
Phase 2: Rip-up and reroute iteration;
Update UC of all the edges;
Build a Queue Q and enqueue all the conflicts;
while Q not empty or not reach Max #Iter. do
    Dequeue(Q) and find all the nets N causing this conflict;
    for each net j in N do
        Dijkstra's algorithm to find cost of reroute;
    end
    Pick net in N with max Δcost as rip-up net;
    Update UC for rip-up net;
    Dijkstra's algorithm to reroute the rip-up net;
    Update UC for reroute net;
    if Reroute causes routing conflict then
        Update HC for corresponding routing resource;
    end
end
Phase 3: Post-processing;
while Prohibited anti-parallel line-ends exists do
    if Line end extension allowed then
        Extend line end to resolve this violation;
    else
        Rip-up and reroute the net causing this violation;
    end
end
```

Algorithm 1: Overall routing algorithm

route of net i. However, the values of α and β used to calculate usage cost are extremely small such that multiple optimal solution can be differentiated.

At the end of each iteration, we update the history cost for all the congested routing resources.

The second phase is negotiated congestion based rip-up and reroute iteration. At each iteration, one routing conflict is resolved. We choose one of the nets causing this routing conflict as the rip-up net. In our implementation, we find out that the choice of rip-up net really affects the final routing solution. Therefore, we try every candidate and find the cost of reroute for each of them. We choose the one that can save the most cost after reroute as the rip-up net and update usage cost due to the removal of the rip-up net. This strategy helps to reduce the total number of rip-up and reroute iteration and generate 0 conflict routing solution, as shown in Section 5. After that, the Dijkstra's algorithm is performed to compute the new path for the rip-up net and usage cost is updated due to the route of the new path. In addition, if the new path causes any conflicts with previously routed net, we update the history cost for the corresponding congested routing resource.

The last phase is post-processing. As mentioned in Section 2, prohibited anti-parallel line-ends violates the design rule of cut mask. In this phase, all the prohibited anti-parallel line-ends are firstly detected. Two options are potentially available to resolve this violation. One is line end extension [9], the other is rip-up and reroute. We prefer the first option if the extension is allowed. If not, we will rip-up and reroute the net causing prohibited anti-parallel line-ends. In our experiments, we find that all the prohibited anti-parallel line-ends can be resolved by line end extension. Finally, a detailed routing solution without design rule violation for SIM type SADP process is generated.

5. EXPERIMENTAL RESULTS

We implemented our algorithm in C/C++ programming language. We run all the experiments on a machine with an Intel Core i5 2.66GHz CPU and 4GB of memory. To the best of our knowledge, this is the first work which systematically considers SIM type SADP/SAQP during detailed routing stage. In order to demonstrate the effectiveness of our proposed algorithm, we compare with the state-of-art detailed routing algorithm which consider LELE [3] and SID type SADP [5] respectively. Testcases C1-C4 and T1-T4 are provided by [3]. For testcases C1-C4, each pin only covers one grid points while each pin might cover several grid points (usually 2-5 grid points) in testcases T1-T4. Testcases Test1-Test10 are provided by [5]. For testcases Test1-Test5, each pin only has a single pin candidate location while each pin has multiple pin candidate locations in testcases Test6-Test10. Four routing layers are assumed and all nets are two-pin nets in all testcases. In Section 5.1, [3] is compared with our algorithm under the assumption of restricted detailed routing, then experimental results for the two types of DPL are discussed. After that, SIM type SAQP detailed routing are performed on the testcase suite, and experimental results are reported. In Section 5.2, [5] is compared with our algorithm, then pros and cons of the two types of SADP are discussed. The experimental resutls are listed in Table 1 and Table 2, where "#C" reports the combined number of design rule violation and detailed routing violation, "#NPJ" gives the number of non-preferred jogs., and "OL" gives the total length of side overlay. Note that the edge costs in our graph model and parameters in the cost function introduced in Section 4.4 are kept same when performing experiments for all the testcases.

5.1 Compared with Seong-I Lei et al. [3]

The motivation to compare our algorithm with [3] is that both of the works adopted the idea of color pre-assignment. In [3], each routing track is assigned one of the two colors and adjacent tracks in horizontal (or vertical) direction are assigned with different colors. The idea greatly reduces coloring conflicts in LELE. However, there are several intrinsic differences between these two types of DPL. LELE allows using stitches to resolve coloring conflicts where SADP does not. Furthermore, we need to consider non-preferred jog minimization during detailed routing stage which will further restrict our solution space. However, LELE has a major disadvantage that it has worse overlay control due to the easy misalignment of two masks. Table 1 shows the performance between our algorithm and [3]. Compared with LELE approach, our approach can generate detailed routing solution with almost same quality in terms of total wirelength and via count. Furthermore, both of algorithms return 0 conflict. The non-preferred jog count can be minimized by our algorithm to a very small number. [3] uses a 3GHz Linux machine with 64 GB memory. We have more than 7X speedup in terms of runtime.

We also implement SIM type SAQP detailed routing and run it on this benchmark suite. As shown in Columns 13 to 17 of Table 1, our detailed routing solution of SAQP is comparable with that of SADP. We achieve better quality in terms of total wirelength, via count, non-preferred jog count, and runtime. Meanwhile, no conflict is reported in the final solution.

5.2 Compared with Lou-Jen Liu et al. [5]

The motivation to compare with [5] is that we want to further study the performance between these two popular types of SADP lithography. [5] is a state-of-art overlay-aware detailed router for SID type of SADP lithography using the cut process

Table 1: Comparison with Seong-I Lei et al. [3]

			Seong-I Lei et al. [3]				Our SADP detailed routing					Our SAQP detailed routing				
Testcases	#Net	Grid size	WL	#Via	#C	CPU(s)	WL	#Via	#C	#NPJ	CPU(s)	WL	#Via	#C	#NPJ	CPU(s)
C1	1500	100×100	9054	3900	0	11	9310	4302	0	4	3	9276	4188	0	5	2
C2	10000	300×300	60325	23036	0	60	61523	24070	0	21	32	61561	23162	0	19	29
C3	1927	400×400	64347	4084	0	42	64379	4214	0	2	11	64369	4046	0	1	13
C4	2400	400×400	64331	5124	0	42	64369	5182	0	1	11	64371	5088	0	0	11
T1	869	500×500	99146	2092	0	122	98975	1978	0	0	53	98977	1956	0	0	51
T2	1036	600×600	128429	2480	0	819	128295	2340	0	1	30	128287	2320	0	0	29
T3	1763	800×800	215035	4060	0	539	214760	3924	0	1	67	214748	3870	0	0	49
T4	3017	1000×1000	194716	6306	0	159	193993	6116	0	1	21	193875	6004	0	0	18
Average			104422.9	6585.3	0	224.3	104450.5	6517.7	0	3.9	28.5	104433.0	6329.2	0	3.1	25.6
Normalized			1.00	0.97		7.87	1.00	1.00		1.00	1.00	1.00	0.97		0.79	0.89

Table 2: Comparison with Lou-Jen Liu et al. [5]

			Lou-Jen Liu et al. [5]					Ours (without non-preferred jog minimization)						Ours (with non-preferred jog minimization)					
Testcases	#Net	Grid size	WL	#Via	Rout.(%)	OL(Pct)	CPU(s)	WL	#Via	Rout.(%)	OL	#NPJ jog	CPU(s)	WL	#Via	Rout.(%)	OL	#NPJ	CPU(s)
Test1	1000	240×240	4610	26	95.3	104(2.3%)	0.4	4124	32	100	0	721	1.2	4220	48	100	0	461	0.1
Test2	1800	340×340	7318	30	96.7	134(1.8%)	1.6	6099	38	100	0	1099	1.2	6553	48	100	0	707	0.2
Test3	4000	400×400	12115	64	97.2	268(2.2%)	6.0	12004	158	100	0	3048	7.4	11972	178	100	0	1409	2.7
Test4	8000	600×600	26745	136	97.4	503(1.9%)	23.1	25316	302	100	0	5627	23.4	25778	324	100	0	1409	2.7
Test5	12000	900×900	40204	136	98.3	424(1.1%)	56.2	37414	232	100	0	8704	19.3	38388	266	100	0	2929	7.4
Test6	1000	240×240	5198	60	96.1	207(4.0%)	0.3	3779	8	100	0	530	0.2	4597	20	100	0	62	0.2
Test7	1800	340×340	9108	64	96.9	259(2.8%)	1.2	6539	10	100	0	940	0.4	8066	22	100	0	64	0.5
Test8	4000	400×400	17379	130	95.6	603(3.5%)	5.4	13193	90	100	0	1989	1.4	15981	146	100	0	196	1.7
Test9	8000	600×600	33589	240	96.2	1031(3.1%)	24.5	32733	34	100	0	5796	7.4	32929	254	100	0	2978	7.2
Test10	12000	900×900	51051	268	97.8	1008(2.0%)	54.8	37970	54	100	0	5932	7.3	46845	124	100	0	289	10.6
Average			20731.7	115.4	96.8	454.1(2.5%)	17.3	17917.1	95.8	100	0	3438.6	6.9	19532.9	143.0	100	0	1337.3	4.0
Normalized			1.16	1.21	0.97		2.51	1.00	1.00	1.00		1.00	1.00	1.09	1.49	1.00		0.39	0.58

which provides higher design flexibility. However, it inevitably introduces side overlay whenever a section of a feature boundary is not surrounded by spacer. In contrast, our approach does not introduce any side overlay since all features are formed by spacers. Table 2 compares the performance of our algorithm and [5]. [5] does not assume a preferred routing direction for each layer and it does not consider sm-jog minimization. Thus, we remove the penalty cost for both non-preferred direction routing and non-preferred jog in our program and run it. Compared with [5], our algorithm reduces total wirelength by 16% and via count by 21%. Furthermore, our algorithm achieves 100% routability while [5] still cannot route all the net successfully for all the testcases. These unroutable nets will further enlarge the difference of total wirelength and via count between our algorithm and [5]. Finally, although [5] tries to minimize the amount of side overlay, it cannot completely eliminate it. On average 2.5% of total wirelength is generated with side overlay error. On the other hand, our approach produces no side overlay error at all. This greatly improves the yield and reduces circuit performance variability. [5] is not able to release the binary, so we have to compare the runtime of two algorithms on different machines. [5] uses a 2.93 GHz Linux work station with 48GB memory. We have more than 2.5X speedup.

Since we did not consider non-preferred jog minimization in above experiments, the non-preferred jog count in the routing solution is high. In order to minimize the non-preferred jog count and improve the yield, we simply add back the penalty cost for non-preferred jog and rerun the program. Columns 15 to 20 in Table 2 show the experimental results. The non-preferred jog count can be effectively reduced by more than 60% with a little overhead on total wirelength and via count. The total wirelength increases by 9% and via count on average increases 47.2. Meanwhile, 100% routability is kept and more than 40% runtime reduction is achieved.

Due to the space limitation, we do not report the SAQP detailed routing solution for this set of testcases. However, the experimental results are similar to those in Section 5.1.

6. CONCLUSION

In this paper, we proposed a detailed routing algorithm for SIM type SADP/SAQP lithography. To the best of our knowledge, this is the first time to systematically consider SIM during detailed routing stage. We pre-assign color to routing panels and restrict mandrel patterns within colored panels. It significantly reduces design rule violation in the final detailed routing solution. Experimental results demonstrate that our proposed algorithm can generate better detailed routing solution for SIM type SADP lithography in terms of total wirelength, via count, routability, and runtime. Meanwhile, no design rule violation is reported. Besides, our approach does not produce any side overlay error. This gives us evidence that SIM type is a promising approach for sub-20nm technology nodes. For the future work, we will consider SID type SADP/SAQP detailed routing problem with similar color pre-assignment idea.

7. ACKNOWLEDGMENTS

This work was supported in part by the Ministry of Science and Technology under grant MOST 103-2220-E-007-002.

8. REFERENCES

[1] Yijian Chen et al. Technological merits, process complexity, and cost analysis of self-aligned multiple patterning. In *Proc. of SPIE*, volume 8326, page 832620, 2012.

[2] Hongbo Zhang et al. Characterization and decomposition of self-aligned quadruple patterning friendly layout. In *Proc. of SPIE*, volume 8326, page 83260F, 2012.

[3] Seong-I Lei, Chris Chu, and Wai-Kei Mak. Double patterning-aware detailed routing with mask usage balancing. In *Proc. of ISQED*, pages 219–223, 2014.

[4] Yuelin Du et al. Spacer-is-dielectric-compliant detailed routing for self-aligned double patterning lithography. In *Proc. of DAC*, pages 1–6, 2013.

[5] Lou-Jen Liu et al. Overlay-aware detailed routing for self-aligned double patterning lithography using the cut process. In *Proc. of DAC*, pages 1–6, 2014.

[6] Jhih-Rong Gao and David Z. Pan. Flexible self-aligned double patterning aware detailed routing with prescribed layout planning. In *Proc. of ISPD*, pages 25–32, 2012.

[7] Chikaaki Kodama et al. Self-aligned double and quadruple patterning-aware grid routing with hotspots control. In *Proc. of ASP-DAC*, pages 267–272, 2013.

[8] Yuelin Du et al. Enhanced spacer-is-dielectric decomposition flow with model-based verification. In *Proc. of SPIE*, volume 8684, page 86840D, 2013.

[9] Yixiao Ding, Chris Chu, and Wai-Kei Mak. Throughput optimization for SADP and e-beam based manufacturing of 1D layout. In *Proc. of DAC*, pages 1–6, 2014.