DOPPLER: DPL-aware and OPC-friendly Gridless Detailed Routing with Mask Density Balancing

Yen-Hung Lin*†, Yong-Chan Ban†, David Z. Pan†, and Yih-Lang Li*
*Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
†Department of Electrical and Computer Engineering, University of Texas at Austin, Texas, USA

ABSTRACT

The printed image of a layout that satisfies the double patterning lithograph (DPL) constraints may not have good fidelity if the layout neglects optical proximity correction (OPC). Simultaneously considering DPL and OPC becomes necessary when generating layouts, especially in routing stage. Moreover, one decomposed design with balanced mask density has a lower edge placement error (EPE) than an unbalanced one [6]. This work proposes a *comprehensive conflict graph* (CCG) to enable detailed routers to simultaneously consider DPL, OPC, and mask density to generate litho-friendly layouts. This work then develops an *DPL-aware and OPC-friendly gridless detailed routing* (DOPPLER) by applying CCG in a gridless routing model. A *density variation threshold annealing-based routing flow* is also proposed to prevent DOPPLER from falling into a sub-optimal mask density balance. Compared with existing DPL-aware detailed routing works, DOPPLER demonstrates an average 73.84% of EPE hotspot reduction with a satisfactory mask density at the cost of an average increase of 0.08% wirelength, 15.14% number of stitches, and 77.28% runtime.

1. INTRODUCTION

The increasing gap between the 193*nm* wavelength used in current lithography and the patterning requirement used in advanced process requires the development of next-generation lithography (NGL). Double patterning lithography (DPL) becomes a feasible means of increasing the pitch size and further enhancing the resolution and depth of focus by decomposing a single layer into two masks [1, 2]. DPL requires *layout decomposition* assigning two features in one layer to opposite *colors* (masks) if their spacing is less than the *minimum coloring spacing*, sp_{dp}. Some layout configuration causes decomposition failure owing to coloring constraints. Generating *stitches* by dividing one feature into at least two parts may resolve decomposition conflicts. However, stitches are highly sensitive to *overlay error*. If possible, one requires minimizing the number of stitches and making the overlapped length of two masks for each stitch exceed the *overlay margin*.

Resolution enhanced techniques (RET) [3] such as optical proximity correction (OPC) are demanded to improve printability even after adopting DPL due to litho-unfriendly patterns such as line-end, jog-corner, etc. However, the increasing manufacturing gap disallows RET alone during mask synthesis [4]. Similar to DPL, physical design automation especially in routing stage needs to consider lithographic proximity effects to generate layouts with satisfactory printability.

Previous works of DPL can be classified into those only decomposing layouts and those generating or modifying the layouts. Kahng *et al*. [5] utilized *conflict graph* to decompose the layout in the post-routing stage. Yang *et al*. [6] balanced the two decomposed masks by applying min-cut partitioning to enhance printability. However, limited by the approaches in post-routing stage such as described above [5, 6], a layout neglecting DPL obstructs the layout decomposition, and redesigning infeasible layouts requires considerable effort. Considering DPL in layout synthesis, especially in detailed routing stage, can bring the promising effect to further decomposition. Cho *et al*. [7] developed the first grid-based detailed routing approach that greedily determined the masks of routed wire segments to avoid generating unresolvable layouts. Lin and Li [8] developed a deferred coloring assignment-based gridless detailed routing flow to escape from the suboptimum that may be reached by the greedy coloring assignment. Yuan and Pan [9] spread wires to simultaneously minimize the number of conflicts and stitches, while introducing as less layout perturbation as possible.

Meanwhile, a layout neglecting lithographic proximity effects may affect the printability of the layout, requiring layout redesign. Several studies have considered lithographic proximity effects in the routing stage to avoid repetition of the time-consuming redesign and simulation. Cho *et al*. [10] proposed a litho-friendly detailed router with weak grid types based on a compact post-OPC litho-metric. Duo *et al*. [4] combined data learning and hotspot prediction techniques to develop compact kernel models and applied the kernels at routing stage to derive litho-friendly interconnect synthesis.

To generate litho-friendly layout after the decomposition of DPL, balancing the densities of two masks under global and local density constraints reduces the edge placement error (EPE) [6]. Limited by the post routing process, they generated stitches to split features to satisfy both global and local density constraints. But there is no work so far to simultaneously consider DPL conflict/stitch minimization and mask density balancing together in routing stage. Moreover, a gridless routing model has higher flexibility to deal with variable routing rules than a grid-based model, and can generate on-grid routing wires with an on-grid feature to satisfy the need for regular layout design. The contribution of this work is to effectively reduce EPE hotspots after mask synthesis with the following novel routing features.

- A *comprehensive conflict graph* (CCG) is proposed to enable a detailed router to consider DPL and OPC simultaneously.
- We present an *DPL-aware and OPC-friendly gridless detailed routing* (DOPPLER) by adopting CCG in a gridless routing model.
- A *density variation threshold annealing-based routing flow* of DOPPLER with *printability-aware PMT evaluation*, where PMT is the routing unit of the gridless routing model, is proposed to generate litho-friendly layouts.

The rest of this paper is organized as follows. Section 2 presents the basic concept and the motivation for this work and formulates the

- Yen-Hung Lin is currently a visiting Ph.D. student at University of Texas at Austin.
- This work was partially supported by the National Science Council (NSC) of Taiwan by Grant NSC 99-2220-E-009 -012, NSC 100-2220-E-009 -046, NSC 100-2917-I-009-006, National Science Foundation (NSF) CCF-0644316, and IBM Faculty Award.

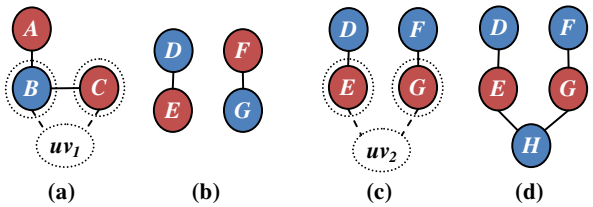


Figure 1. Routing-friendly characteristics of ICG: (a) constant-time conflict cycle detection; (b) two ICSGs before routing; (c) an undetermined vertex uv_2 merges these two ICSGs; (d) the result of lazy ICG update of this routing.

problem. Sections 3 and 4 present CCG and the proposed methods for DOPPLER, respectively. Section 5 summarizes the experimental results. Finally, Section 6 draws conclusions.

2. PRELIMINARIES AND MOTIVATION

2.1. Innovative Conflict Graph

Lin and Li [8] proposed an innovative conflict graph (ICG) to enable detailed routers to consider DPL. ICG contains *determined* and *undetermined vertices* to represent routed and routing wire segments, respectively. An edge in ICG indicates the spacing of the wire segments represented by its terminal vertices is smaller than sp_{dp}. Notably, there are *determined* and *undetermined edges* in ICG to represent the DPL relation between routed/routing and routed/routed wire segments. Each determined vertex is assigned a *pseudo color* to represent the potential color. An ICG comprises several *innovative conflict subgraphs* (ICSGs), and two ICSGs have no determined and undetermined edges between them. An odd length cycle in ICG indicates a coloring conflict.

Three routing-friendly characteristics of ICG assist detailed routing in detecting DPL conflicts during propagation. 1.) *Constant-time conflict cycle detection* confirms that any coloring conflict occurs according to the states of connected determined vertices. In Fig. 1(a), inserting undetermined vertex uv_1 causes one coloring conflict because determined vertices *B* and *C* belong to the same ICSG and their pseudo colors differ. In Fig. 1, the solid/dashed lines represent determined/undetermined edges. 2.) *Lazy ICG update* accelerates the routing process by postponing the real ICG update until obtaining the final result for each routing. Figures 1(b)-(d) show an example of a lazy ICG update. Figure 1(b) displays two ICSGs with assigned pseudo colors. An undetermined vertex uv_2 merges two ICSGs by connecting determined vertices *E* and *G* whose pseudo colors differ. To maintain the correctness of further conflict cycle detection, the ICSG containing *G* flips its pseudo colors, as shown in Fig. 1(c). After obtaining the final routing result, in which uv_2 is selected as a routed wire segment, the ICG is updated as shown in Fig. 1(d). 3.) *Light-weight routing overhead* enables detailed routers to record some of the data rather than a complete conflict graph in each propagation to reduce the routing overhead.

2.2. Routing Models

NEMO [11, 12] is a new implicit connection graph-based gridless router. NEMO deals with tile propagation on a non-uniform grid map. For each net, NEMO constructs a routing graph. Before the routing graph construction, NEMO expands each obstacle and routed net by half of a wire width hw_w and one wire spacing sp_w to generate contours. NEMO constructs the implicit connection graph by extracting all borders of contours. In the propagation stage, NEMO performs *path propagation* by identifying adjacent *pseudo-maximum horizontally/vertically stripped tiles* (PMTs) of

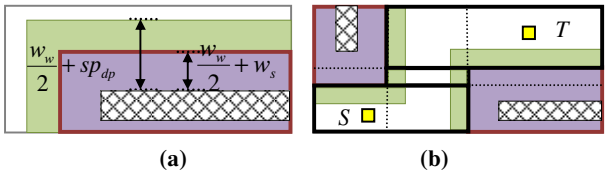


Figure 2. Routing graph construction in DPLAG [9]: (a) contour and DPL-shadow generation; (b) routing graph construction and DPL-aware PMT extraction.

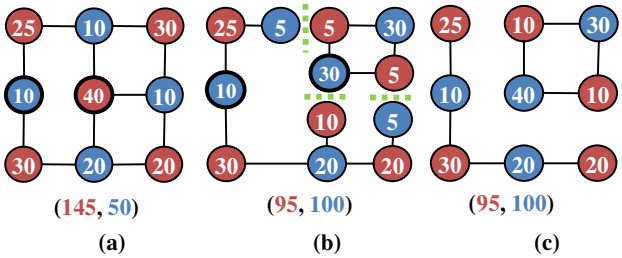


Figure 3. Effect of density constraint on layout configuration presented by ICG: (a) layout that ignores density constraint; (b) generating three stitches to balance density in post-routing stage; (c) density-driven layout balances the density without generating any stitch.

the last PMT in the minimum-cost path and then expanding the connected PMT list. The path propagation is repeated until the PMT containing the target is reached. Accordingly, NEMO generates routing wire segments by retracing the routing result and then places new wire segments on the layout. To allow for DPL effect between routed and routing wire segments, DPLAG [8] constructs *DPL-shadows* by expanding routed wire segments by hw_w+sp_{dp} to represent the affected DPL region of routed wire segments. Figure 2 presents an example of routing graph construction. In Fig. 2(a), the routed net is expanded by hw_w+sp_w/hw_w+sp_{dp} to generate a contour/DPL-shadow. The routing graph is then constructed as the dashed lines in Fig. 2(b), in which three PMTs with black bold borders are passed from *S* to *T*, and are traced by DPLAG to determine the real path. When one routing wire segment passes through a DPL-shadow, DPLAG generates one undetermined edge between the undetermined vertex and the determined vertex on the routed wire of which the DPL-shadow is imposed.

2.3. Motivation

Balancing the densities of the two masks in decomposition improves printability [6]. A layout neglecting density constraints during routing may require generating unnecessary stitches to balance the density in the post-routing stage. Figure 3 shows the effect of the density constraint on layout configuration represented by an ICG. Figure 3(a) depicts a layout represented by ICG ignoring the density constraint. In Fig. 3(b), generating three stitches improves the density, at the cost of yield loss. Clearly, a density-driven layout in Fig. 3(c) requires no stitches to balance the density, but achieves the same quality as in Fig. 3(b) (with almost the same metal area in the two layers).

Yang *et al.* [6] considered not only the global density, but also the local density to improve printability. However, satisfying the local density constraint may still impair printability. In Fig. 4, the layout configuration satisfies the DPL constraint, and wire segments *A* and *E* are of the same mask. However, *A* and *E* are litho-unfriendly patterns which impedes OPC to insert features to diminish the image variation [10]. Assigning A and E to different

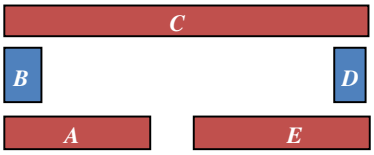


Figure 4. Necessity of explicitly considering OPC constraint in DPL where *A* and *E* form a litho-unfriendly pattern.

masks benefits their printed images, resulting in violating DPL constraints of the original layout. Therefore, explicitly considering OPC together with the local density constraint can greatly enhance printability without causing additional DPL violations. The OPC constraint requires that two features are assigned to different colors for further OPC because they are litho-unfriendly patterns, even though their spacing exceeds sp_{dp}.

Density and OPC constraints must be simultaneously considered in the DPL-aware routing process. However, balancing the density may prevent satisfaction of the OPC constraint. In Fig. 3(a), the two determined vertices with bold borders must satisfy the OPC constraint. To satisfy the density constraint, one of them must be split into two segments that have different colors, potentially resulting in the OPC constraint violation because two determined vertices with bold borders are assigned to the same mask, as shown in Fig. 3(b). Therefore, a detailed router must resolve the conflicts between density and OPC constraints when DPL constraints are satisfied.

2.4. Problem Formulation

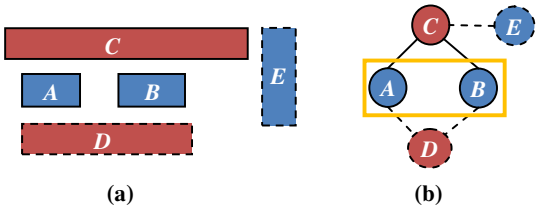
DPL-aware and OPC-friendly gridless detailed routing problem: Given a netlist, a OPC design rule set DR^{OPC}*, and a density variation threshold* $T_{den} \in [0,1]$ *detailed routing and coloring assignment are performed to minimize the number of stitches, DPL constraint violations, and OPC constraint violations under the density constraint.*

3. COMPREHENSIVE CONFLICT GRAPH

Neglecting OPC effects in layout synthesis may inhibit OPC even when the layout is decomposable. Considering DPL and OPC simultaneously in detailed routing can significantly reduce the design turn-around time. ICG [8] prohibits from generating coloring conflicts in propagation stage, dramatically limiting the solution space when directly handling OPC constraints as DPL constraints. This work proposes a *comprehensive conflict graph* (CCG) based on ICG to enable detailed routers to consider DPL, OPC, and mask density constraints simultaneously. CCG is defined below.

Comprehensive conflict graph (CCG): An CCG $G^{CC} = (V^D, E^D, C^P_A, V^U, E^U, IP, W)$, where V^D denotes the determined vertex set; E^D denotes the determined edge set; C^P_A denotes the pseudo color set of all (both determined and undetermined) vertices ($C^P_A = \{B, R\}$); V^U denotes the *undetermined vertex* set; E^U denotes the undetermined edge set; *IP* denotes the *interference pair* set; and *W* denotes the vertex weight set. An interference pair, $ip = (v_1, v_2)$, is a pair of two vertices, v_1 and v_2, between which the minimum spacing exceeds sp_{dp} and they are litho-unfriendly patterns. A vertex weight represents the wire length of one determined vertex. An CCG may comprise several connected components, and each component is a subgraph of CCG and named as CCSG.

Conflicting interference pair: An interference pair compels two vertices in the same interference pair to be given different pseudo colors. An interference pair is said to be conflicting if its two vertices cannot be given different pseudo colors, such that the OPC constraint is violated.



(a) (b)

Figure 5. Example of CCG presentation: (a) layout of one layer where solid/dashed border rectangles are routed/routing wires, and *A* and *B* form a litho-unfriendly pattern; (b) CCG of (a) where solid/dashed border circles are determined /undetermined vertices, solid/dashed edges are determined /undetermined edges, and the rectangle containing *A* and *B* represents that *A* and *B* form an interference pair.

CCG has the following OPC-friendly properties:

Property 1. If the separation between two vertices is less than sp_{dp} and they are litho-unfriendly patterns, an edge in CCG, connects these two vertices, then these two vertices will be assigned to two different pseudo colors to satisfy the DPL constraint. The OPC constraint between these two vertices is thus satisfied as well.

Property 2. If the separation between two vertices is larger than sp_{dp} but they are litho-unfriendly patterns, then these two vertices have no edge connecting them in CCG but they form an interference pair. The two vertices of a non-conflicting interference pair are given different pseudo colors, i.e., the OPC constraint is satisfied.

Property 3. The two vertices of a conflicting interference pair are bad for printability, since these two vertices are assigned to the same pseudo color (mask).

Figure 5(a) displays the layout of one layer, and Fig. 5(b) depicts the CCG of the layout in Fig. 5(a). Notably, A and B are litho-unfriendly, and the corresponding vertices in CGG are a conflicting interference pair, as shown in Fig. 5(b). The following subsections describe two OPC-friendly characteristics of CCG: *OPC constraint violation detection*, and *density-driven CCG update*. Finally, the routing-friendly characteristic of CCG is introduced.

3.1. OPC Constraint Violation Detection

CCG provides interference pairs to detect OPC constraint violations in the DPL-aware routing process when no DPL constraint violation occurs. CCG detects an OPC constraint violation in path propagation using the following lemmas.

Lemma 1. *When n new DPL relations are established between an undetermined vertex uv and n CCSGs, such as* sg_i*, for* $1 \le i \le n$*, the current propagation merges n CCSGs by connecting them to uv. An existing interference pair* (v_1, v_2)*,* $v_1 \in sg_i$*,* $v_2 \in sg_j$ *and* $i \ne j$*, becomes conflicting if the path from* v_1*–uv–*v_2 *has an even length.*

Lemma 2. *If the current propagation generates a new undetermined vertex uv and a new interference pair (uv, v), where* $v \in sg$*, and if uv connects to sg by a new DPL relation and the path from v to uv has even length, then the interference pair (uv, v) is conflicting.*

Due to the space limitation, the proofs are omitted.

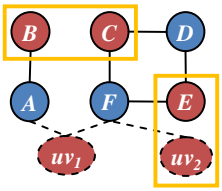


Figure 6. Example of OPC constraint violation detection

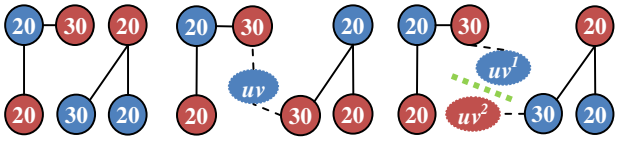


Figure 7. Density-driven CCG update: (a) two CCSGs; (b) merging two CCSGs by *uv* may cause mask density constraint violation; (c) generating one stitch by splitting *uv* into uv^1 and uv^2 can satisfy the mask density constraint.

Figure 6 shows an example of OPC constraint violation detectio with an CCG that has six determined vertices, five determined edges, and one interference pair (B, C), where uv_1 and uv_2 are two undetermined vertices associated with two propagations. Before uv_1 is inserted, the interference pair (B, C) is non-conflicting because B and C are in different CCSGs. Two CCSGs are merged by connecting uv_1 to A and F using undetermined edges. Since the path from B to C has even length, the pseudo colors of A and F are identical, making the interference pair (B, C) conflicting according to *Lemma 1*. The interference pair (B, C) becomes conflicting according to *Lemma 1*. The next propagation forms an undetermined vertex uv_2, which connects to F by one undetermined edge, and then an interference pair (uv_2, E) is formed bcause the separation between uv_2 and E is larger than sp_{dp} but they are litho-unfriendly. Also, the new edge that connects uv_2 to F forms a new path of even length from uv_2 to E, producing a new conflicting interference pair (uv_2, E) accoring to *Lemma 2*. According to *Lemma 2*, (uv_2, E) is a conflicing interference pair. Each propagation dynamically updates *IP*, the interference pair set. The number of OPC constraint violations is the total number of conflicting interference pairs.

3.2. Density-Driven CCG Update

Rather than recording entire modified CCG during routing propagation, maintaining the correct pseudo colors is the most important task in detecting DPL and OPC conflicts, even after several CCSGs are merged. The lazy ICG update [8] flips pseudo colors of some merged ICSGs when the pseudo colors of connected determined vertices differ. However, direct pseudo color flipping may violate the density constraint. For each routing, an *annealing density variation threshold*, t_{den}, is computed as the allowable variation between two colors in one layer as the density constraint. The following section will introduce the density variation computation. Figure 7(a) displays two CCSGs with total wire lengths of determined vertices in two pseudo colors. In Fig. 7(b), merging two CCSGs using an undetermined vertex, such as uv, requires flipping the pseudo colors of one CCSG for further conflict detection, subsequently increasing the density variation. Hence, the density constraint may be violated. To satisfy the mask density constraint, splitting uv into uv^1 and uv^2 by generating one stitch, if possible, reduces the mask density variation after merging CCSGs, as shown in Fig. 7(c). Merging two CCSGs using an undetermined vertex, such as uv, may require flipping the pseudo colors of one CCSG for further conflict detection, subsequently increasing the density variation. Hence, the density constraint may be violated. To satisfy the mask density constraint, splitting uv into uv^1 and uv^2 by generating one stitch, if possible, reduces the mask density variation after merging CCSGs. Vice versa, when one undetermined vertex merges several CCSGs by connecting to determined vertices with the same pseudo color, flipping the pseudo colors of some CCSGs by generating stitches at the undetermined vertex may also reduce the density variation.

During propagation, merging n CCSGs, such as sg_1 to sg_n, may require color flipping to maintain the correctness of the pseudo colors. The density variation of CCG after n CCSGs are merged, var_{den}, is computed as follows.

$$var_{den} = \frac{\left| \sum_{1\le i\le n} B(sg_i) + \sum_{sg\in UM} B(sg) - \left(\sum_{1\le i\le n} R(sg_i) + \sum_{sg\in UM} R(sg) \right) \right|}{\sum_{1\le i\le n} B(sg_i) + \sum_{1\le i\le n} R(sg_i) + \sum_{sg\in UM} B(sg) + \sum_{sg\in UM} R(sg)} \quad (1),$$

where UM is the unmerged CCSG set, and $B(sg_i)$ and $R(sg_i)$ represent the total wire length of routed wire segments represented by the determined vertices of the CCSG sg_i that are assigned to pseudo colors B and R, respectively. Notably, the wire length of routing wire segments (undetermined vertices) is neglected here because the stitch positions are undecided in path propagation. Assume, without loss of generality, that the total wire length of routed wire segments in CCG that are assigned to pseudo color B exceeds that of vertices assigned to pseudo color R. All CCSGs in UM are sorted in increasing order of $(R(sg) - B(sg))$. Assume $|UM| = m$, sg_{n+i}, $1 \le i \le m$, is the sorted sequence of CCSGs in UM, and sg_{n+k} is the first CCSG in the sorted sequence of UM for which $(R(sg) - B(sg))$ exceeds zero. If var_{den} exceeds t_{den}, for $l \ge k$, the pseudo colors of the CCSG sg_{n+l} in UM are sequentially flipped to decrease the density variation until var_{den} is smaller than t_{den}. If var_{den} still exceeds T_d after flipping the pseudo colors of the unmerged CCSG sg, whose $(R(sg) - B(sg)) \ge 0$, then the merged CCSG is split into $(m+1)$ CCSGs by generating m stitches at the undetermined vertex, and the permutation assignment of pseudo colors among $(m+1)$ CCSGs is found greedily to satisfy the density constraint.

The CCG update is postponed until obtaining the final routing result of each routing: the undetermined vertices are inserted into CCG as determined vertices; the undetermined edges are replaced by determined edges, and an interference pair set is dynamically maintained to prevent an OPC constraint violation.

3.3. Routing-Friendly Characteristic

CCG effectively enables detailed routers to consider DPL, OPC, and density constraints simultaneously in the routing process to improve printability because CCG inherits the light-weight routing overhead of ICG. As well as recording the merged CCSGs, connected determined vertices, and information about pseudo color flipping, each propagation records the following data. For OPC constraint violation detection, CCG requires each propagation to record the updated interference pair set for further propagation. Thus, CCG can correctly detect conflicting interference pairs. For mask density balance, CCG requires each propagation to record information about coloring flipping for all CCSGs, including flipped unmerged CCSGs. Therefore, CCG can correctly compute the mask density variation.

4. DOPPLER

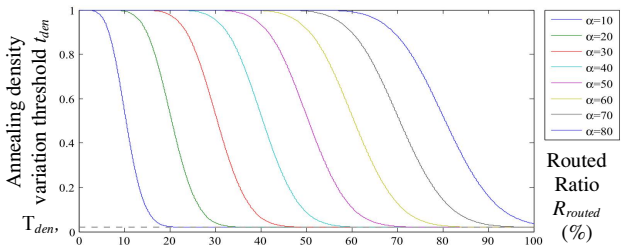


Figure 8. Annealing density variation threshold versus routed ratio.

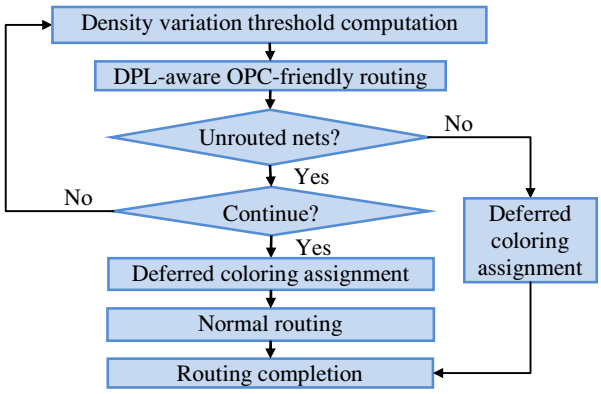


Figure 9. Overview of density variation threshold annealing-based routing flow.

This work develops *DPL-aware and OPC-friendly gridless detailed routing* (DOPPLER) which concentrates on accomplishing detailed routing for all nets and generating a decomposable routing outcome with satisfatory printability after OPC. The proposed DOPPLER is applied to the gridless routing model of NEMO [11] to consider DPL, OPC, and mask density constraints simultaneously by applying the proposed *DPL-aware OPC-friendly routing scheme*. This work presents an approach to apply CCG to the NEMO gridless model, and a *density variation threshold annealing-based routing flow* to balance the mask density variation. With the assistance of CCG, DOPPLER can consider DPL, OPC, and mask density constraints to evaluate the routing cost.

4.1. Density Variation Threshold Annealing-Based Routing Flow

DOPPLER balances the density of two masks in the routing process to enhance the printability. If the density variation threshold is too strict in early stage, the routing result may fall into a suboptimal density variation. Moreover, DOPPLER can reduce the difference between two masks by inserting stitches to maintain or flip the pseudo colors of CCSGs. This work proposes a density variation threshold annealing-based routing flow to escape a local optimum owing to the strict density constraint. Before each routing, the density variation threshold t_{den} is dynamically computed based on the routed ratio, $R_{routed} \in [0, 100]$, as follows.

$$t_{den}(R_{routed}) = \mathrm{T}_{den} + (1 - \mathrm{T}_{den}) \times \frac{\Gamma(\alpha, R_{routed})}{\Gamma(\alpha, 0)}, \quad (2)$$

where $\Gamma(a,b) = \int_b^{\infty} t^{a-1} e^{-t} dt$ is the normalized upper incomplete gamma function of a and b, and T_{den} is the given density variation threshold. Figure 8 plots the curve of annealing density variation threshold versus routed ratio for various α. When the routed ratio approaches 100%, the annealing density variation threshold rapidly approaches T_{den}. In this study, α is set to 70.

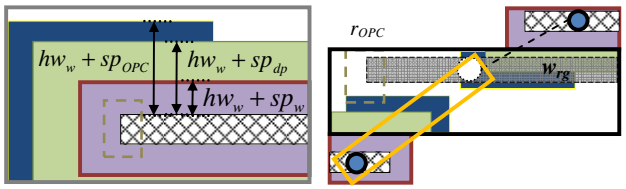


Figure 10. OPC-shadows and path propagation of CCG-inside gridless routing: (a) OPC-shadow construction; (b) path propagation in PMT with DPL-shadows and OPC-shadows.

Figure 9 shows the proposed density variation threshold annealing-based routing flow. The design principle of DPL-aware OPC-friendly routing is to avoid DPL and OPC constraint violations and to satisfy the mask density constraint, computed by the density variation threshold computation. In this stage, each routed segment w_{rd} examines its neighboring wire segments and records those that are too close to w_{rd}, such that the DPL and OPC constraints must be satisfied. The colors of routed nets have not yet been determined. Until all nets are routed or the DPL-aware OPC-friendly routing cannot converge, the deferred coloring assignment [8] determines the colors of routed nets directly from pseudo colors in CCG. If unrouted nets exist, normal routing is used and greedily determines colors of routing nets simultaneously.

4.2. DPL-Aware OPC-Friendly Routing

4.2.1. CCG-Inside Gridless Routing Model

This work develops an CCG-inside gridless routing model to consider DPL, OPC, and density constraints in detailed routing. DOPPLER constructs *OPC-shadows* to represent the affected region of OPC by routed wire segments. Notably, the affected region must be computed dynamically using the given DR^{OPC} of routed and routing wire segments in a layout. DR^{OPC} such as the litho-metric [10] determines the spacing of various patterns. Interference pair generation in the proposed routing model is introduced below. DOPPLER then expands routed wire segments by $hw_w + sp_{OPC}$ to construct OPC-shadows where sp_{OPC} is the OPC spacing and computed dynamically, as shown in Fig. 10(a). Notably, an RET-shadow of one routed wire segment excludes its DPL-shadow. In each PMT, the configurations of the routing wire segments can be predicted with the aid of PMT [11]. Therefore, the OPC affected region of routing wire segment is inserted dynamically for OPC conflict detection. One determined and one undetermined vertices form an interference pair if one of the following two conditions is met.

1.) One routing wire segment or its OPC affected region passes only through the OPC-shadow of one routed wire segment w_{rd}, and do not overlap the DPL-shadow of w_{rd}.

2.) The OPC affected region of the routing wire segment passes through the DPL-shadow of one routed wire segment w_{rd}, and may also pass through the OPC-shadow of w_{rd}.

Figure 10(b) shows one routing wire segment w_{rg} (the gray dashed-border rectangle) passing through the DPL-shadow of one routed wire segment and its OPC affected region r_{OPC} (the blank dashed-border rectangle) passing through the OPC-shadow of the other routed wire segment (Condition 1). Therefore, one undetermined edge and one interference pair are generated to

Algorithm: Printability-Aware PMT evaluation
Input: A PMT (*PMT*), an annealing density variation threshold (t_{den}), DPL-shadows (*SH*), interference pair set (*IP*) and potential wire segments passing *PMT*.
Output: Evaluated routing cost.
begin
1. compute density variation var_{den};
2. ***if*** ($var_{den} > t_{den}$) flip unmerged CCSGs to satisfy t_{den};
3. ***if*** ($var_{den} > t_{den}$) {
4. tentatively generate *m* stitches in routing wire segment and flip (*m*+1) CCSGs greedily to decrease var_{den};
5. ***if*** ($var_{den} > t_{den}$) generate one density violation;
6. ***else*** realize *m* stitches; }
7. compute DPL constraint violation penalty based on *SH*;
8. ***for*** (each pair *ip* in *IP*) {
9. ***if*** (vertices of *ip* are in the same CCSG and have identical pseudo color) {
10. report all even-length paths between *ip*;
11. ***if*** (all even-length paths are breakable)
12. generate sufficient stitches;
13. ***else*** detect one OPC constraint violation; } }
14. compute routing cost;
end

Figure 11. Algorithm of printability-aware PMT evaluation.

maintain the DPL and OPC constraints between the undetermined vertex associated with w_{rg} and the corresponding determined vertices, respectively. Therefore, OPC and DPL conflicts can be detected in the routing process.

4.2.2. Printability-Aware PMT Evaluation

In the path propagation stage, DOPPLER utilizes DPL- and OPC-shadows and CCG to evaluate the routing cost to improve printaility of layouts after OPC. Figure 11 displays the algorithm for printability-aware PMT evaluation. PMT is the basic routing unit in the adopted routing model. Each propagation explores a new PMT. For the current PMT, DOPPLER determines whether an DPL or OPC constraint violation is caused when the potential routing wire segment passes through the PMT. When a DPL/RET constraint violation occurs, DOPPLER attempts to break the odd-length/even-length cycle/path by generating one stitch at the cost of yield loss or by enlarging the spacing to expel the constraint. Moreover, to satisfy the mask density constraint, DOPPLER may generate stitches to reduce the density difference between wire segments of two colors (lines 1–6).

DOPPLER treats routing wire segments as undetermined vertices. DOPPLER attempts to balance the mask density during propagation. If the annealing density variation threshold t_{den} cannot be satisfied ever after generaing stitches, one density violation is generated. For the currently explored PMT, DOPPLER records the set of determined vertices whose DPL-shadows are passed by the routing wire segments, as *SH*. Since the undetermined vertex will connect to the determined vertices in *SH* with undetermined edges, DOPPLER queries each pair of determined vertices in *SH* to determine whether the two vertices of this pair are connected by an odd-length path. If so, then an odd-length cycle (DPL constraint violation) is generated after connecting the undetermined vertex to the determined vertices in *SH*. This detection is very simply performed by verifying the pseudo colors of each pair of determined vertices in *SH*, in a mannar similar to that described elsewhere [8] (line 7). To detect an OPC constraint violaiton, DOPPLER detects the status of each existing interference pair with *Lemma 1* and *Lemma 2*. Practically, the path length can also be determined by simply examining the pseudo colors of two vertices (lines 8–9). DOPPLER attempts to solve each conflicting interference pair by breaking all even-length paths between the two vertices of the pair at the cost of additional stitches (lines 10-12). If the conflicting between one interference pair is unsolvable, then one OPC constraint violation is detected (line 13). The penalty cost is the computed based on the density violation, stithch number, DPL and OPC constraint violations (line 14).

Table 1. Statistics of benchmarks

Circuit	Size (μm^2)	#Layer	#Net	#Conn.	#Pins
s5378	43.5×23.9	3	1694	3124	4818
s9234	40.4×22.5	3	1486	2774	4260
s13207	66.0×36.5	3	3781	6995	10776
s15850	70.5×38.9	3	4472	8321	12793
s38417	114.4×61.9	3	11309	21035	32344
s38584	129.5×67.2	3	14754	25556	42931

5. EXPERIMENTAL RESULTS

The algorithm herein is implemented in C++ language on a workstation with 8-core 3GHz CPU with 48GB memory. This work adopts six benchmarks with original minimum feature size 360*nm*. We scale all benchmarks by a factor of 0.1× to approach 32*nm*. The statistics of benchmarks are shown in Table 1. The DPL parameters are set by referring to [5] – 8*nm* for overlay margin and 58*nm* for minimum coloring spacing, such as sp_{dp}. Instead of adopting an OPC design rule set, we simply set the OPC spacing sp_{OPC} as 81*nm*, which exceeds sp_{dp} by more than half of the wire width, to demonstrate the effectiveness of the proposed algorithm. The density variation threshold is set to be 2% by referring to [6]. To analyze edge placement error (EPE), we used Calibre-Workbench to get lithographic printed images for 45*nm* node test patterns. Our optical parameters are wavelength λ = 193*nm*, numerical aperture (NA) = 0.92 dry lithography, and dipole unpolarized illumination σ = 0.9/0.7. The thicknesses of the photo-resist (PR) and the bottom anti-reflecting coating (BARC) are 150*nm* and 38*nm*, respectively. For analyzing the EPE (edge placement error) in our test case, we used systematic lithography process variation, such as focus ±50*nm* and dose ±5%.

No previous works focus on considering lithographic proximity and DPL simultaneously in routing stage. We compare DOPPLER to DPLAG [8], which is a DPL-aware gridless router with the goal to minimize stitches and unresolvable conflict cycles (UCC), to demonstrate the necessity of considering DPL and lithographic proximity simultaneously in routing stage. Table 2 compares routing results of DPLAG and DOPPLER, including wirelength, stitches number, UCC number, density variation of all layers, normalized EPE hotspots, and runtimes. Neither DPLAG nor DOPPLER generates UCC. DOPPLER controls the density variation for all cases to satisfy the given threshold, demonstrating that the proposed density variation threshold annealing-based routing flow can effectively control the density variation. Compared to DPLAG which neglects the lithographic proximity, DOPPLER reduces the EPE hotspots from 72.67% to 75.75% (on average 73.84%) at the cost of an average increase of 0.08% wirelength, 15.14% stitches, and 77.28% runtime.

Table 3 further shows the EPE hotspots comparison of the two masks of each layer between DPLAG and DOPPLER. Compared to the litho-unaware router (DPLAG), DOPPLER effectively reduces the EPE hotspots from 54.55% to 95.15%. The EPE hotspots reduction of DOPPLER in Metal1 is mostly less than that in Metal2 be-

cause the pins of all cases locate in Metal1, restricting DOPPLER to prevent generating litho-unfriendly layouts in Metal1. On the contrary, the EPE hotspots reduction in Metal3 is better than those in Metal1 and Metal2 because there is more space for DOPPLER to generate litho-friendly layouts.

6. CONCLUSION

OPC can improve the printability of original layouts. A DPL-decomposable routing result that neglects lithographic proximity effect may impede further RET such as OPC. Moreover, a balanced decomposed design has less edge placement error than an unbalanced one. To realize considering lithographic proximity and DPL in routing stage, this work proposes CCG to enable detailed routers to consider OPC and DPL constraints together. It develops the first DPL-aware and OPC-friendly gridless detailed router (DOPPLER) with a density variation threshold annealing-based routing flow to generate litho-friendly layouts. Experimental results demonstrate that DOPPLER can effectively yield layouts with less EPE hotspots, at the cost of little increase in wirelength, stitches number, and runtime.

7. REFERENCES

[1] J. Huckabay *et al*., "Process Results Using Automatic Pitch Decomposition and Double Patterning Technology (DPT) at k1eff < 0.20," *Proc. SPIE Conf. on Photomask Technology*, pp.634910-1 - 634910-11, 2006.

[2] J. Park *et al*., "Application Challenges with Double Patterning Technology (DPT) Beyond 45*nm*," *Proc. SPIE Conf. on Photomask Technology*, pp.634922-1 - 634922-12, 2006.

[3] F. M. Schellenberg, "Resolution Enhancement Technology: The Past, the Present and Extensions for the Future," *Proc. SPIE Conf. on Optical Microlithography XVII*, pp. 1-20, May 2004.

[4] D. Ding, J.-R. Gao, K. Yuan, D. Z. Pan, "AENEID: A Generic Lithography-friendly Detailed Router Based on Post-RET Data Learning & Hotspot Prediction," *DAC*, 2011.

[5] A. B. Kahng, C.-H. Park. X. Xu, and H. Yao, "Layout Decomposition for Double Patterning Lithography," *ICCAD*, pp.465-472, 2008.

[6] J.-S. Yang *et al*., "A New Graph-Theoretic, Multi-Objective Layout Decomposition Framework for Double Patterning Lithography," *ASP-DAC*, pp. 637-644, Jan. 2010.

[7] M. Cho, Y. Ban, and D. Z. Pan, "Double Patterning Technology Friendly Detailed Routing," *ICCAD*, pp.506-511, Nov. 2008.

[8] Y.-H. Lin, Y.-H. Li, "Double Patterning Lithography Aware Gridless Detailed Routing with Innovative Conflict Graph," *DAC*, pp. 398-403, 2010.

[9] K. Yuan, D. Z. Pan, "WISDOM: Wire Spreading Enhanced Decomposition of Masks in Double Patterning Lithography," *ICCAD*, pp. 32-38, 2010.

[10] M. Cho, K. Yuan, Y. Ban, D. Z. Pan, "ELIAD: Efficient Lithography Aware Detailed Router with Compact Post-OPC Printability Prediction," *DAC*, pp. 504-509, 2008.

[11] T.-C. Chen, G.-W. Liao, Y.-W. Chang, "Predictive Formulae for OPC with Applications to lithography-friendly routing," *TCAD*, vol. 29, no. 1, Jan. 2010.

[12] Y.-L. Li, X.-Y. Chen, and Z.-D. Lin, "NEMO: A New Implicit Connection Graph-based Gridless Router with Multilayer Planes and Pseudo-tile Propagation," *TCAD*, vol.26, no.4, pp.705-718, Apr. 2007.

[13] Y.-N. Chang *et al*., "Non-Slicing Floorplanning-based Crosstalk Reduction on Gridless Track Assignment for a Gridless Routing System with Fast Pseudo-Tile Extraction," *ISPD*, pp.134-141, 2008.

Table 2. Routing Result Comparison between DPLAG (DG) [8] and DOPPLER (DR).

Circuit	Wirelength (*nm*)		# Stitches		# UCC		Density Variation (%)		Normalized EPE Hotspots		Runtime (s.)	
	DG	DR	DG	DR	DG	DR	DG	DR	DG	DR	DG	DR
s5378	7530	7538	27	29	0	0	11.40	0.64	1	0.2425	10.79	21.28
s9234	5633	5642	21	24	0	0	13.95	1.74	1	0.2542	9.81	12.09
s13207	17985	17988	87	98	0	0	13.39	1.63	1	0.2601	70.73	113.44
s15850	22353	22355	52	66	0	0	10.85	1.66	1	0.2733	72.66	112.44
s38417	48940	48973	195	237	0	0	17.47	1.52	1	0.2666	293.36	593.79
s38584	63694	63754	250	270	0	0	11.89	1.93	1	0.2728	308.20	662.78
Ave.	1	1.0008	1	1.1514			1	0.1175	1	0.2616	1	1.7728

Table 3. EPE Hotspots Comparison between DPLAG (DG) [8] and DOPPLER (DR).

Circuit	M1-Mask1		M1-Mask2		M2-Mask1		M2-Mask2		M3-Mask1		M3-Mask2	
	DG	DR *red.* (%)	DG	DR *red.* (%)	DG	DR *red.* (%)	DG	DR *red.* (%)	DG	DR *red.* (%)	DG	DR *red.* (%)
s5378	1962	71.13	3406	74.31	1042	82.25	1473	80.04	33	54.55	93	92.47
s9234	1764	74.66	2757	70.00	856	88.67	1228	74.92	16	81.25	36	72.22
s13207	4868	72.66	7366	71.84	2055	82.04	3138	75.33	103	76.70	185	80.54
s15850	5063	69.88	8820	71.64	2342	79.59	3670	72.92	113	80.53	227	95.15
s38417	12251	72.00	20721	71.01	5281	82.11	8107	75.13	210	80.48	456	78.07
s38584	16282	71.23	26872	71.38	6139	78.56	10100	75.00	328	73.78	553	74.86