O(n) Layout-Coloring for Multiple-Patterning Lithography and Conflict-Removal Using Compaction

(Invited Paper)

Rani S. Ghaida*, Kanak B. Agarwal§, Sani R. Nassif§, Xin Yuan†, Lars W. Liebmann‡, Puneet Gupta*
* UCLA, Electrical Engineering Dept.
§ IBM Corp., Austin Research Lab
† IBM Corp., San Jose
‡ IBM Corp., Semiconductor Research & Development Center
{rani, puneet}@ee.ucla.edu {kba, nassif, xinyuan, lliebman}@us.ibm.com

Abstract—[1] **The use of multiple patterning optical lithography for sub-20nm technologies has become inevitable with delays in adopting the next generation of lithography systems. The biggest technical challenge of multiple patterning is failure to reach a manufacturable layout-coloring solution, especially in dense layouts. This paper offers a post-layout solution for the removal of conflicts, i.e., patterns that cannot be colored without violations. The proposed method consists of three steps essentially: layout coloring, exposure layers and geometric rules definition, and, finally, layout legalization using compaction and multiple-patterning rules as constraints. The method is general and can be used for different multiple-patterning technologies including LELE double-patterning (DP), tripe/multiple-patterning (i.e., multiple litho-etch steps), and self-aligned double patterning (SADP). For demonstration purposes, we apply the proposed method in this paper to remove conflicts in DP. We offer an O(n) layout-coloring algorithm for DP, which is up to 200X faster than the ILP-based approach, and extend it for multiple-patterning (MP). The conflict-removal problem is formulated as a linear program (LP), which permits an extremely fast run-time (less than 1 minute in real time for macro layout). The method was tested on cells from a commercial 22nm library designed without any multiple-patterning awareness; for many cells, the method removes all conflicts without any area increase; for some complex cells, the method still removes all conflicts but with a modest 6.7% average increase in area.**

I. Introduction

One of the most favorable solutions for extending optical lithography for future nodes is double/multiple-patterning (DP/MP). This paper focuses on MP in a multiple litho-etch steps process (LELE). Hereafter, the term MP is used to denote MP in multiple LELE steps and the term DP is used to denote LELE DP.

For a layout to be MP-compatible, layout features that violate the minimum spacing of single patterning must be assigned to different masks. MP-mask assignment is essentially a color-labeling problem [3] and will be referred to as layout-coloring hereafter. The difference from the labeling problem of graph theory is that a layout polygon can be a composite of the layouts of the different masks. The splitting of polygons into multiple parts is known as stitching and the location where different masks join is called a stitch. Although stitching complicates the labeling problem and stitches may be a cause for yield loss, stitching is needed to conform many problematic layout patterns to MP without the need for layout modification (by breaking odd cycles in the conflict graph for DP). Even with stitching, the coloring of many patterns may not be possible without violating the minimum same-color spacing. Such patterns are called native MP conflicts and resolving these conflicts – with certain layout perturbation – is the biggest challenge facing the deployment of MP technologies.

A. Prior art in DP coloring and conflict removal

DP coloring has been the subject of extensive research [4–8] in recent years, while little work has been done on MP coloring and these will be discussed later on in Section II. Prior works in DP coloring [6–8] still suffer from some limitations, however. One limitation common to all previous works is that they perform segmentation of the layout into rectangles prior to the coloring and this segmentation has many drawbacks. First, it complicates the problem as it forces the consideration a lot of extra stitch locations that should never be used (see Figure 1). The second drawback of segmentation is that it forces the method to use a single spacing-rule value. Because rectangles are mapped into nodes, it is hard to distinguish between side-to-side, tip-to-side, and tip-to-tip spacing design rules (DRs), which have different values in modern technologies.

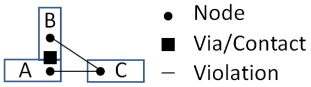


Figure 1. Illustration of the drawbacks of layout segmentation into rectangles used in prior works. Here, segmentation effectively introduces a candidate-stitch location for the coloring at the joint of A and B, which should never used because both A and B have spacing violation with C. In layout legalization, segmentation forces A as well as B to entirely cover the via, which is unnecessary and may impact the layout area.

Some prior works focus on the removal of MP-conflicts with automated layout perturbation [9–11]. These works generally formulate the problem as an ILP (except [11]), which is excessively time-consuming to solve. Moreover, they do not perform a layout legalization in a general context and, therefore, the removal of one conflict may create new conflicts in other parts of layout or design-rule violations at other layers. Previous works in this area also uses layout-segmentation, which lead to some drawbacks in addition than the ones discussed earlier. The problem is further complicated because the automated layout perturbation solver (ILP or compaction) needs to maintain the connectivity of rectangles at joints (e.g., L-shape) through additional constraints. Moreover, because the constraints of the solver are defined between rectangles, overlap rules with features from the top and bottom layers cannot be handled correctly (see Figure 1).

B. Our approach

This work offers a framework for MP conflict removal for standard cells and macros. We follow a different approach for the coloring than prior works. Essentially, we use DR-dependent projection to determine the violating features that may cause DP conflicts and their actual, possibly non-rectangular, shapes. We then formulate the problem as a labeling problem of violating features that we solve in a $O(n)$ algorithm. In our method, all candidate stitches that may be useful are considered, which guarantees reaching a conflict-free coloring solution when one exist. Using a linear program (LP), MP conflicts are removed and the layout is legalized simultaneously across multiple layers by edge-based layout perturbation. This layout legalization is performed through layout compaction formulated as a *minimum perturbation problem*[2]. The proposed methodology allows the layout designer to design with conventional single-patterning layers and DRs, masking him from the complexity in dealing with double-patterning layers and requirements.

An overview of the flow of our methodology to achieve MP-compatible layouts is depicted in Figure 2. Using existing non MP-compatible layouts designed with conventional rules, we perform an optional step of layout simplification at MP layers for the possible

[1] This paper is based primarily on [1], which describes the work in detail, and is supplemented with the work on MP coloring presented in [2].

[2] Unlike [11] that uses minimum-area metric for compaction. The advantages of minimum layout perturbation metric over the minimum area metric for layout compaction are discussed in [12, 13].

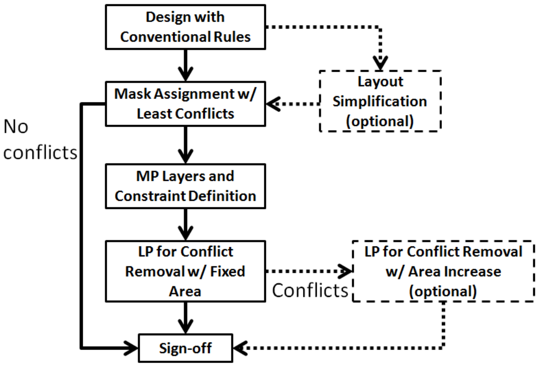

Design with Conventional Rules
Layout Simplification (optional)
Mask Assignment w/ Least Conflicts
No conflicts
MP Layers and Constraint Definition
LP for Conflict Removal w/ Fixed Area
Conflicts
LP for Conflict Removal w/ Area Increase (optional)
Sign-off

Figure 2. The flow for our proposed method to achieve MP-enabled layout design.

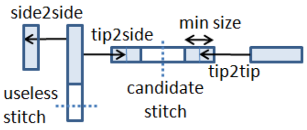


Figure 3. DR-dependent projection to identify violating parts and stitch locations. Violating parts are the blue features and non-violating parts are the clear features.

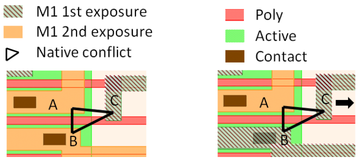
sacrifice of non-critical parts as described in Section III-B. We then carry out MP coloring while considering all candidate locations. If the layout is uncolorable, we first define MP layers and geometric constraints and apply LP-based compaction to remove conflicts and legalize the layout simultaneously across all layers while minimizing layout perturbation. During this legalization the layout area is constrained and, in case some native conflicts remain unresolved, the legalization is optionally repeated with unconstrained area.

II. MP Layout-Coloring

In this section, we describe our O(n) method for DP layout-coloring and its extension for MP. We follow a different approach for the MP layout-coloring than what is presented in the literature. We first find all parts of the layout that have DP spacing violations with neighboring features and, then, we color these violating parts. In this way, candidate locations of stitches are automatically defined and can be easily minimized as we show later in this section. In the end, non-violating parts can be labeled with any color. If a non-violating part touches features of the same color, we label it with the same color to avoid introducing extra stitches; whereas, if a non-violating part touches features of different colors, we label it with the different colors of the touching features to maximize the overlap region of the masks. The details of this implementation follow.

A. O(n) Layout-Coloring for DP

We start with DR-dependent projection to identify violating parts as illustrated in Figure 3. From each edge in the layout, side or tip, we determine features in its neighborhood that violate the corresponding spacing DR. Unlike previous works that can only allow a single spacing rule, we consider three spacing DRs: side-to-side (S2S), tip-to-side (T2S), and tip-to-tip (T2T). Violating parts that are smaller than the minimum feature size are grown to meet the requirement. Projection prior to the layout-coloring allows the automatic detection of all useful candidate stitch locations. Stitches are beneficial if and only if they are located in non-violating parts that separate two or more violating parts. A stitch introduced inside any violating part will automatically create a conflict with the neighboring feature that created the violation. The same fact holds for stitches in small non-violating parts that cannot guarantee the minimum overlap length of masks and they are disregarded (by joining the attached violating parts). Moreover, a stitch in any non-violating part that is attached to a single violating part is useless since the entire region of such non-violating part can always be colored with the same color as the attached violating part.



(a) (b)

Figure 4. Layout-coloring of features of an odd cycle can affect the efficiency of conflict removal. In (a), the conflict is on M1 between shapes A and B and can only be fixed if the gates are spaced apart and area is increased; in (b), the conflict is on M1 between shapes B and C and can be fixed by moving C in the direction of the arrow without increasing area.

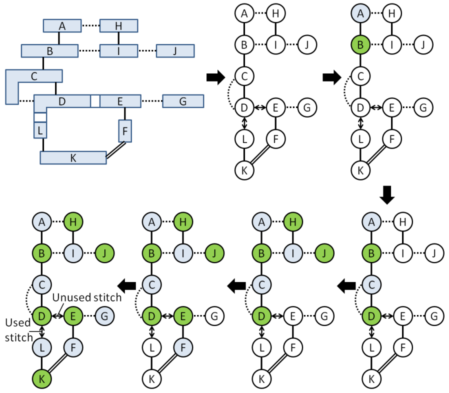


Figure 5. An illustrating example showing each step of our layout-coloring process for an isolated region of the layout.

The main objective of layout-coloring is to reach a solution with the minimum number of conflicts. A secondary objective is to minimize stitches, which may incur yield loss due to overlay errors between the different exposures. Because stitches can remove certain conflicts, we consider all possible stitch locations during layout-coloring and get rid of stitches that do not affect the number of conflicts.

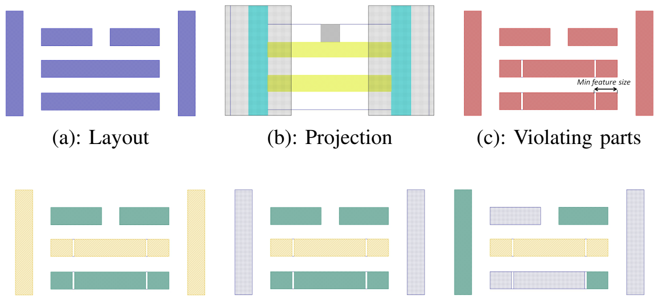
Although an odd cycle will always result in a DP conflict no matter the coloring, deciding what features go on the same color can affect the efficiency of the conflict removal (see Figure 4). To take advantage of this observation, we design our layout-coloring algorithm to handle and be aware of the criticality of violations.

After DR-dependent projection, the coloring problem is reduced to the coloring of violating parts. Our proposed $O(n)$ coloring algorithm, where n is the number of violations and candidate stitches, is illustrated through the example of Figure 5. We start by constructing the conflict graph, where violating parts are represented by nodes and violations and stitches are represented by arcs. We then identify connected components (in $O(n)$) and, for each connected component, we pick a violation-arc with preference to more critical violations and color its two connected nodes with different colors. Whenever a new node is colored, its connected arcs get added to first-in-first-out queues of the different types of violations and stitches to be processed next. A new arc (possibly a stitch-arc) is picked from the different queues with preference to arcs corresponding to more critical violations and to violation-arcs over stitch-arcs. This process is repeated until all arcs in the component are processed. Each node is colored only once: when a violation-arc is processed, the two nodes are colored differently and, when a stitch-arc is processed, the two nodes are colored with the same color.

It is important to note that we further reduce the number of used stitches after a coloring solution has been reached by possibly flipping the colors of certain violating parts. The coloring of an entire subcomponent, defined as portion of a component where all nodes are connected with violating-arcs, is flipped whenever the number of used stitches is reduced. A greedy approach is followed to do this flipping with more-beneficial flips performed first.

B. Extending the Layout-Coloring for MP

The layout-coloring problem for MP is much more complicated than layout-coloring for DP. First, while determining whether the



(a): Layout (b): Projection (c): Violating parts

(d): C0/C1 coloring (e): C0/C2 coloring (f): C1/C2 coloring

Figure 6. Example illustrating the application the different steps involved in our methodology for MP layout-coloring with three colors: C0, C1, and C2. In (b), T2S violations are expanded and the parts of the target layout that are covered after the expansion are the T2S violating parts. T2T and S2S violating parts are identified similarly.

layout is DP-compatible is achieved in polynomial time (by odd-cycle detection in the conflict graph), determining whether the layout is MP-compatible is a NP-complete problem even for triple-patterning. Second, because additional colors are available, identifying candidate stitch locations prior to the coloring is not possible as in the case of DP. In MP, a stitch can be between any pair of the available colors and, consequently, stitch locations are color-dependent and candidate stitches can be located almost anywhere in the layout.

The only complete prior work that covers triple-patterning (TP) layout-coloring for lines in LELELE process is the work of [14]. The biggest limitation of the work of in [14] is that it uses candidate stitches that are only DP candidate stitches. In reality however, many patterns require the insertion of TP stitches for them to be colorable without violations as in the example of Figure 6.

We propose a novel methodology for MP layout-coloring that leverages MP stitching capability. Rather than simplifying the MP stitching problem by using DP candidate stitches only (as in previous works), the methodology considers additional MP candidate stitch locations to give coloring higher flexibility to resolve coloring conflicts. The key idea is to distinguish violating parts based on violated rules, i.e. S2S, T2S, and T2T, and insert candidate stitch locations (in addition to DP stitches) in S2S violating parts at the interface with the other types of violating parts. The intuition for this is that S2S violating parts are typically large, so they may accommodate more candidate stitches. Also, the only place where a candidate stitch is absolutely useless is when it is inserted at a location that is violating with two other shapes that are also violating between one another; clearly, this occur much less often in the case of S2S violating parts than in the case of T2S or T2T violating parts.

To deal with MP coloring complexity, the methodology employs multiple DP coloring steps, which leverages existing infrastructure developed for DP layout-coloring. Hence, MP coloring can be performed in $O(n)$ using our DP coloring algorithm described earlier. The MP layout-coloring methodology is illustrated through the example of Figure 5 and further details can be found in our previous work of [2].

III. Removal of Coloring-Conflicts Using Compaction

After the MP layout-coloring is complete, our objective is to make the layout compatible with MP and resolve the conflicts while minimizing layout perturbation. In this section, we describe our approach for the removal of coloring-conflict using compaction with minimum layout perturbation and layout pre-processing methods for more efficient conflict removal.

A. *Conflict Removal With Compaction and Min Layout Perturbation*

We use the method proposed in [12] for layout legalization with minimum perturbation as the objective. The layout is represented as a constraint graph where nodes correspond to the layout edges and arcs correspond to the DRs that need to be met between any two layout edges. Arcs are assigned weights that correspond to the values of rules as illustrated in Figure 7. Layer-to-layer connectivity is maintained through the DRs between the layers, which are represented in the graph by arcs between nodes of the different layers.

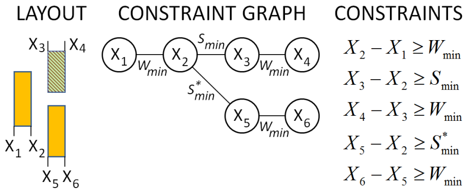

LAYOUT
CONSTRAINT GRAPH
CONSTRAINTS
X3
X4
X1
X2
X5
X6
W_{min}
S_{min}
S^*_{min}
W_{min}
W_{min}
$X_2 - X_1 \geq W_{\min}$
$X_3 - X_2 \geq S_{\min}$
$X_4 - X_3 \geq W_{\min}$
$X_5 - X_2 \geq S^*_{\min}$
$X_6 - X_5 \geq W_{\min}$

Figure 7. Example of x-direction constraint graph construction and constraint definition for a MP layer. W_{min} is the minimum width rule, S_{min} is the side-to-side spacing rule in the layout, and S^*_{min} is the side-to-side same-color spacing rule.

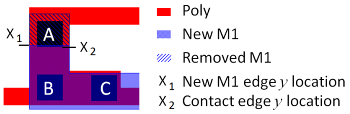


Figure 8. Illustration of M1 simplification for possible sacrifice of redundant contacts.

The same-color layouts of any MP layer are defined as stand-alone layers. Spacing DRs between features of the same color including S2S, T2S, and T2T are mapped into arcs between the nodes of the stand-alone same-color layers in the constraint graph. DRs that define the interaction between the different-color layouts of MP (e.g., minimum overlap length) are mapped into arcs between the nodes of the stand-alone same-color layers. For the interactions across different layers in the stack (e.g., M1 and contacts), we define any MP layer as the union of its different-color layouts and map across-layers DRs into arcs between nodes of the union layers[3].

B. *Pre-processing of Layouts for More Efficient Conflict Removal*

In actual layouts, we observe that some DP conflicts can be avoided by simple notch removal prior to coloring. In addition, many conflicts on the M1 layer are caused by segments that are added to cover redundant contacts/vias or to maximize the pin-access region. Redundant contacts/vias improve manufacturability, but they are *not absolutely required*. The same is true for pin segments that are used only to maximize the access region and, consequently, improve the routing efficiency. We take advantage of these observations and, as an option, we pre-process the layout *before the layout-coloring* to allow the *possible* sacrifice of redundancy and extra pin segments to improve the results of the DP conflict removal framework.

If a metal layer is formed with MP, the line-end part of the metal that covers a redundant contact (or via) is removed, as shown in Figure 8, and overlapping the redundant contacts back with metal is specified as a recommended constraint. The LP of the conflict removal method will meet this recommended constraint only when it is possible without creating a MP conflict or any DR violations. In other words, redundant contacts will be sacrificed only when necessary to resolve conflicts. To ensure recommended contacts still get a chance to be covered by metal after the layout is perturbed, we add a required constraint to keep redundant contacts at the same spacing and aligned to the associated required contact. Similarly, M1/M2 pin segments that do not connect to any other layer in the layout stack are removed for possible sacrifice. Again, a recommended constraint is set to allow the recovery of the original pin-shapes without creating violations.

IV. Experimental Setup and Results

In this section, we demonstrate our methodology for MP-enabled design by applying it for DP. DP layout-coloring was implemented using Calibre SVRF code [15] and C++ with OpenAccess database. The DP conflict removal with layout perturbation was implemented and integrated into the minimum perturbation based VLSI artwork legalization system [12].

[3] Rather than using layers of the same-color layouts and encounter the same problem highlighted in Figure 1.

We test our layout-coloring method on the poly-layer layouts presented in [6] and compare the results with the layout-coloring approaches proposed by this previous work (ILP-based). Using the same values for the minimum same-color spacing rule and overlap length rule, we obtain 92× to 233× run-time improvement over the pure ILP approach with a modest 4% to 8.8% increase in the number of stitches (see Table I). Compared with the conflict cycle detection approach of [6], our method is 62× to 223× faster and lead to almost the same number of stitches (ranging from -1% to +4%).

Our MP conflict removal framework was tested for DP on a commercial 22nm standard-cell and macro layouts. We assume M1 is double-patterned and apply the conflict removal method for layouts that have DP conflicts. The M1 minimum spacing rule is 40nm and we use a value of 15nm for the minimum overlap length and 80nm for all minimum same-color spacing rules.

For standard cells, the method removes all DP conflicts without any area increase in five out of nine cells; for the remaining four cells, the method still removes all conflicts with a 6% average increase in area. For macro layouts, the method reduces the number of DP conflicts from 13 to 7 conflicts for one macro and from 53 to 31 conflicts for another without any increase in area. When the area is allowed to increase, the method removes all remaining conflicts with an average area increase of 8.7%. The runtime of the entire flow for the largest macro layout (460 transistors) is less than one minute in real time (< 2 seconds CPU time).

Although the method targets primarily the design-enablement of MP, the method can be used during technology development to evaluate the density impact of different MP technologies and their design rules.

Table I
RESULTS OF OUR DP COLORING AT THE POLY LAYER AND COMPARISON WITH PREVIOUS WORK OF ILP-BASED COLORING AS WELL AS THE CONFLICT CYCLE DETECTION (CCD) BASED APPROACH [6]. "CUTS" REFERS TO STITCHES, "SECS" REFERS TO THE NUMBER OF SECONDS, AND "MIN" REFERS TO THE MINIMUM OVERLAP LENGTH).

		ILP [6]		**CCD [6]**		**Our approach**	
Design	**Min**	**Cuts**	**Secs**	**Cuts**	**Secs**	**Cuts**	**Secs**
ART-A 45(70%)	8	24290	564.6	25521	378.6	25480	6.1
ART-B 45(70%)	10	72828	2887.4	76550	2316.8	76634	20.5
ART-C 45(70%)	8	121916	8291.2	127935	7895.8	126715	35.5
ART-A 45(90%)	13	25432	612	26629	391	27691	6.3
ART-B 45(90%)	10	76292	2892.2	79836	2355.2	82089	20.5
ART-C 45(90%)	8	126238	8129	132303	8205	135558	37.5

Table II
RESULTS OF APPLYING OUR DP CONFLICT REMOVAL METHOD WITH FIXED/UNFIXED AREA ON CELLS AND MACRO LAYOUTS FROM A COMMERCIAL 22NM LIBRARY.

	Original		**w/ Fixed Area**	**w/ Area Increase**	
Cell	**N. Area**	**Conflicts**	**Conflicts**	**Conflicts**	**Area Increase**
LCB	1	**1**	**0**	-	-
latch1	1.6	**3**	**2**	0	9.1%
oai	1.6	**2**	**0**	-	-
scan latch	2.3	**5**	**3**	**0**	6.2%
xor	2.4	**2**	**0**	-	-
latch2	4.3	**19**	**8**	**0**	3.3%
nand4	4.7	**4**	**0**	-	-
latch3	5.3	**4**	**3**	**0**	5.4%
nand3	6.7	**7**	**0**	-	-
LCB ctrl1	13.7	**13**	**7**	**0**	8.3%
LCB ctrl2	50.3	**53**	**31**	**0**	9.1%

V. CONCLUSIONS

We proposed a novel framework to enable MP in the design. Part of the framework is an $O(n)$ algorithm for DP layout-coloring, which guarantees a conflict-free solution if one exists, and a methodology for efficient MP layout-coloring, which leverages MP stitching capability. The automated MP conflict removal and layout legalization are performed simultaneously across all layout layers while minimizing perturbation using a LP. The method enables designing with conventional DRs masking the designer from the complexity in dealing with MP layers and requirements. The way we formulate the problem allowed us to achieve high-quality results with extremely fast runtime (less than 1 minute in real time for a 460-transistor macro). The method targets primarily standard-cell layouts. Complete standard-cell based designs can be formed, as in [16], either by fixing the colors at the cell-boundaries, to ensure that flipping the cell coloring resolves all DP conflicts that may be induced by cell placement, or by using a correct-by-construction approach, where enough spacing is allocated between colored features of the cell and the cell boundary to prevent placement-induced conflicts. The method is not limited to standard cell-based designs, however, and it can also be applied for full-custom layouts and interconnect layers in complete designs. Although we demonstrate the method on LELE DP, the method is more general and can be naturally extended for other MP technologies including LELE MP and SADP. For SADP, all that is needed is a layout-coloring method as [17] and a set of design rules to ensure SADP compatibility of the layout as in [18].

REFERENCES

[1] R. S. Ghaida, K. B. Agarwal, S. R. Nassif, X. Yuan, L. W. Liebmann, and P. Gupta, "A framework for double patterning-enabled design," in *Intl. Conf. on Computer-Aided Design*, Nov. 2011, pp. 14–20.

[2] R. S. Ghaida, K. B. Agarwal, L. W. Liebmann, S. R. Nassif, and P. Gupta, "A novel methodology for triple/multiple-patterning layout decomposition," in *Proc. SPIE*, vol. 8327, 2012, p. 83270M.

[3] M. Drapeau, V. Wiaux, E. Hendrickx, S. Verhaegen, and T. Machida, "Double patterning design split implementation and validation for the 32nm node," in *Proc. SPIE*, vol. 6521, 2007, pp. 652 109–1.

[4] T.-B. Chiou *et al.*, "Development of layout split algorithms and printability evaluation for double patterning technology," in *Proc. SPIE*, vol. 6924, 2008, pp. 69 243M–1.

[5] A. Tritchkov *et al.*, "Double-patterning decomposition, design compliance, and verification algorithms at 32nm HP," in *Proc. SPIE*, vol. 7122, 2008, pp. p. 71 220S–1.

[6] A. B. Kahng, C.-H. Park, X. Xu, and H. Yao, "Layout decomposition approaches for double patterning lithography," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 6, pp. 939–952, 2010.

[7] K. Yuan, J.-S. Yang, and D. Z. Pan, "Double patterning layout decomposition for simultaneous conflict and stitch minimization," in *Intl. Symp. on Physical Design*, 2009, pp. 107–114.

[8] 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 *IEEE Asian and South Pacific Design Automation Conference*, 2010, pp. 637–644.

[9] C. Hsu, Y. Chang, and S. R. Nassif, "Simultaneous layout migration and decomposition for double patterning technology," in *IEEE Intl. Conf. on Computer-Aided Design*, 2009, pp. 595–600.

[10] K. Yuan and D. Z. Pan, "WISDOM: wire spreading enhanced decomposition of masks in double patterning lithography," in *IEEE Intl. Conf. on Computer-Aided Design*, 2009, pp. 32–38.

[11] S.-Y. Chen and Y.-W. Chang, "Native-conflict-aware wire perturbation for double patterning technology," in *IEEE Intl. Conf. on Computer-Aided Design*, 2009, pp. 556–561.

[12] F. L. Heng, Z. Chen, and G. Tellez, "A VLSI artwork legalization technique based on a new criteria of minimum layout perturbation," in *ACM/IEEE Intl. Symp. on Physical Design*, 1997, pp. 116–121.

[13] J. Zhu, F. Fang, and Q. Tang, "Calligrapher: a new layout-migration engine for hard intellectual property libraries," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 9, pp. 1347–1361, 2005.

[14] B. Yu, K. Yuany, B. Zhang, D. Ding, and D. Z. Pan, "Layout decomposition for triple patterning lithography," in *Intl. Conf. on Computer-Aided Design*, 2011, pp. 1–8.

[15] Calibre standard verification rule format manual v2009. [Online]. Available: http://www.mentor.com/

[16] L. Liebmann, D. Pietromonaco, and M. Graf, "Decomposition-aware standard cell design flows to enable double-patterning technology," in *SPIE*, vol. 7974, 2011, p. 79740K.

[17] Y. Ban, A. Miloslavsky, K. Lucas, S.-H. Choi, C.-H. Park, and D. Z. Pan, "Layout decomposition of self-aligned double patterning for 2d random logic patterning," in *Proc. SPIE*, vol. 7974, 2011, p. 79740L.

[18] Y. Ma, J. Sweis, H. Yoshida, Y. Wang, J. Kye, and H. J. Levinson, "Self-aligned double patterning (sadp) compliant design flow," in *Proc. SPIE*, vol. 8327, 2012, p. 832706.