Self-Aligned Double Patterning Aware Pin Access and Standard Cell Layout Co-Optimization

Xiaoqing Xu, *Student Member, IEEE*, Brian Cline, *Member, IEEE*, Greg Yeric, *Senior Member, IEEE*, Bei Yu, *Member, IEEE*, and David Z. Pan, *Fellow, IEEE*

***Abstract*—Self-aligned double patterning (SADP) is being considered for use at the 10-nm technology node and below for routing layers with pitches down to ∼50 nm because it has better line edge roughness and overlay control compared to other multiple patterning candidates. To date, most of the SADP-related literature has focused on enabling SADP-legal routing in physical design tools while few attempts have been made to address the impact SADP routing has on local, standard cell (SC) I/O pin access. At the same time, via layers are used to connect the local SADP routing layers to the I/O pins on lower metal layers. Due to the high via density on the Via-1 layer, the litho-etch-litho-etch (LELE)-aware Via-1 design becomes a necessity to achieve legal pin access at the SC level. In this paper, we present the first study on SADP-aware pin access and layout optimization at the SC level. Accounting for SADP-specific and Via-1 design rules, we propose a coherent framework that uses depth first search, mixed integer linear programming, and backtracking method to enable LELE friendly Via-1 design and simultaneously optimize SADP-based local pin access and within-cell connections. Our experimental results show that, compared with the conventional approach, our framework effectively improves pin access of the SCs and maximizes the pin access flexibility for routing.**

***Index Terms*—Double patterning, pin access, self-aligned double patterning (SADP), standard cell (SC) layout, Via-1 assignment.**

I. Introduction

DUE to the resolution limits of 193-nm photolithography, double patterning techniques, and regular layout have been widely used to extend semiconductor process technology scaling [2]–[4]. The design rules that enable double patterning (color decomposition, forbidden pitches, etc.) are much more restrictive than the basic rules used previously in technology nodes larger than 20 nm. In addition, the expectation to continue Moore's law translates to the same density and area scaling every node. That means, the physical design tools need to access standard cell (SC) input/output (I/O) pins in more congested areas with increasingly restrictive rules.

One way that SC designers can assist physical design tools is through intelligent, optimized SC I/O pin design. Unfortunately, the complex design rules and neighbor interactions that exist due to various multiple patterning techniques like litho-etch-litho-etch (LELE) and self-aligned double patterning (SADP) make human-driven layout almost impossible at 14-nm technologies and below. That means automated SC layout design and optimization are needed to provide flexible I/O pin access.

SADP, in particular, is a viable candidate for lower layer metalization with regular patterns at the 10-nm technology node, due to better overlay and line edge roughness (LER) [5] control compared to LELE. To deploy the SADP technique for routing layers in practical designs, designers need to ensure that layout patterns are SADP-friendly to achieve successful layout decomposition. The SADP layout decomposition problem has been studied, as shown in [6]–[9]. For regular layout, the line-space array decomposition method can efficiently decompose SADP-based geometries and achieve good pattern fidelity and process margin [3], [10]. An example of line-space array decomposition is demonstrated in Fig. 1. Fig. 1(a) shows regular layout on horizontal tracks. We can assign different colors to patterns on neighbor tracks in Fig. 1(b). Fig. 1(c) shows the mandrel mask design and spacer deposition. Then, the trim mask and spacer define the target layout as demonstrated in Fig. 1(d).

To incorporate SADP constraints into early design stages, there are several studies [11]–[14] dealing with the SADP-aware routing problem. However, to date, works studying how multiple patterning and decomposition impact SC I/O pin design are lacking, especially as pin congestion and routability become increasingly critical to the overall physical design results. Since most modern-day SC designs primarily use Metal-1 for local connections and I/O pins, Metal-2 design is essential for SC I/O pin access. References [15] and [16] focused on detailed routing accounting for pin access. Nieberg [15] focused on the gridless routing and proposes efficient algorithms to improve the pin accessibility and routability. Ozdal [16] presented the formulation and solution for the detailed routing for dense pin clusters.

SADP-based Metal-2 wires, in particular, present a new set of problems to SC I/O pin access. Specifically, because of the decomposition of SADP into the mandrel and trim masks, one cannot simply rely on via locations to determine

Manuscript received July 18, 2014; revised October 7, 2014 and December 23, 2014; accepted January 22, 2015. Date of publication February 4, 2015; date of current version April 17, 2015. This work was supported in part by the National Science Foundation (NSF), and in part by Semiconductor Research Corporation (SRC). A preliminary version of this paper was presented at the International Symposium on Physical Design in 2014 [1]. This paper was recommended by Associate Editor M. Ozdal.

X. Xu, B. Yu, and D. Z. Pan are with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78731 USA (e-mail: xiaoqingxu.austin@utexas.edu).

B. Cline and G. Yeric are with ARM Inc., Austin, TX 78735 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2015.2399439

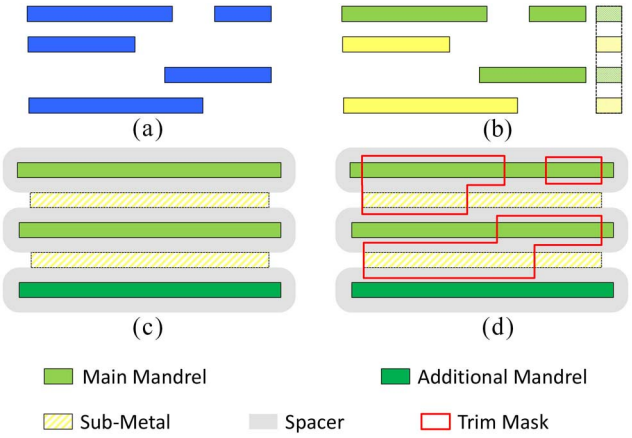

Fig. 1. Line-space array decomposition. (a) Target layout. (b) Layout coloring. (c) Mandrel mask design. (d) Trim mask design.

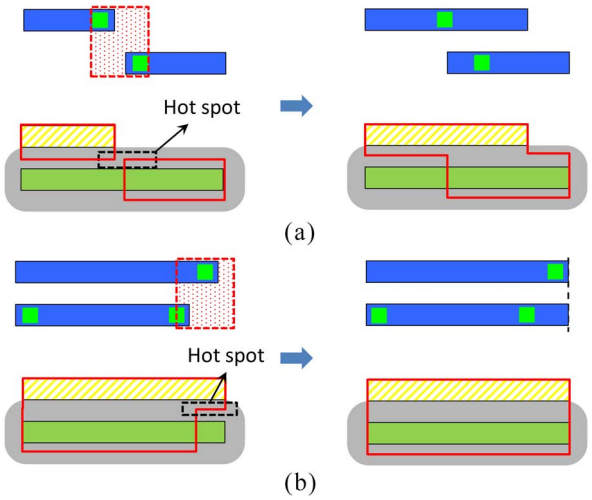

Fig. 2. Line-end extension techniques. (a) Anti-parallel line ends. (b) Parallel line ends.

line-end positions of Metal-2 wires. As shown in Fig. 2, SADP yield can be enhanced by simple line-end extensions that are dependent on both via placement and neighboring wire placement. For example, in Fig. 2(a) and (b), respectively, the extension of anti-parallel line ends and parallel line-end alignment both help to avoid hot spots on the trim masks [17].

In general, the ideal location of geometries is not as straightforward under SADP constraints and is more dependent on the neighborhood around the geometry in question. This means all SADP-based metal designs, including pin access, and within-cell connections have to be optimized simultaneously during the I/O pin design phase. Line-end extension techniques have been deployed to deal with the cut mask manufacturability previously in [18] and [19]. To minimize the effective gaps in the layout, Zhang *et al.* [18] propose a greedy optimal algorithm to extend the line ends of metal wires. Zhang *et al.* [19] presented a mask cost reduction formulation to improve the layout manufacturability with the help from line-end extension techniques. Instead of mask cost reduction, our work seeks trim mask friendly, SC-level pin access design under SADP constraints, *a priori*, before the SC library is ever used in a placed-and-routed block.

Fig. 3. (a) Via-1 patterns. (b) Odd cycle in conflict graph.

In addition, the Via-1 layer is used to connect the Metal-2 wires and Metal-1 I/O pins. Due to the high density of Via-1s in future technology nodes, double patterning (LELE) is needed to manufacture the Via-1 layer. Hence, we need to enable LELE friendly Via-1 design to achieve legal pin access at the SC level. For instance, Fig. 3(a) shows possible Via-1 patterns at the SC level. A solid edge is added between two Via-1s if they are too close to each other and can not be single patterned. A conflict graph can be extracted from the Via-1 patterns, as illustrated in Fig. 3(b). Determining the double patterning decomposability of given layout patterns is a well-studied topic [20]–[24]. If the conflict graph deduced from the given layout patterns does not contain odd cycles, the layout patterns are double patterning (LELE) friendly. It can be clearly observed that an odd cycle, denoted by the red arrow, exists in the conflict graph of Fig. 3(a), which reveals the Via-1 patterns in Fig. 3(a) are not LELE friendly. Odd-cycle free LELE-aware Via-1 design is equally important to determining SC pin access as SADP-aware Metal-2 design.

In this paper, we formulate this issue as an SC I/O pin access problem and illustrate the usefulness of our methodology at the 10-nm technology node. To solve this problem efficiently, this paper proposes a depth first search (DFS)-based method to achieve LELE-friendly Via-1 assignment and a mixed integer linear programming (MILP)-based technique that simultaneously optimizes the Metal-2 wires used for pin access and within-cell connections of SCs. In addition, using the backtracking method, we extend this technique to each pin access strategy and maximize pin accessibility for each cell in the 10-nm library. Our main contributions are summarized as follows.

1) To the best of our knowledge, this is the first work that addresses SC I/O pin access design at the SC level.
2) We present a DFS-based technique to achieve LELE-aware optimal Via-1 assignment for pin access and cell connections.
3) We propose a MILP-based optimization methodology to enable SADP-aware Metal-2 layout design for pin access and within-cell connections.
4) The pin access and cell layout co-optimization (PICO) is proposed to systematically maximize the pin access flexibility for the entire SC library.

The rest of this paper is organized as follows. Section II introduces background material relevant to the pin access design issue. Section III shows the formulation of the SADP-aware pin access design problem, including several definitions and our design target. Section IV presents our DFS-based technique for optimal Via-1 assignment and MILP-based methodology for Metal-2 layout optimization. Section V extends our optimization technique to the entire SC library based on

the backtracking method. Section VI demonstrates the effectiveness of our optimization framework, and compares the SADP-aware pin access optimization (PAO) to the conventional approach that uses basic design rule checks. Compared with [1], we also illustrate the impact that LELE-aware Via-1 assignment has on the final pin access solution. Section VII concludes this paper.

II. Preliminaries

A. Line-Space Array Decomposition

The continued geometric scaling of process technology depends on multiple patterning and increased layout regularity [3]. Thus, at the 10-nm node and beyond, we assume that the Metal-2 layout will be extremely regular. Furthermore, in the 10-nm commercial technology, we used the preferred direction of Metal-2 routing was horizontal. After studying the Metal-2 routing tracks, we made the following observation.

Observation 1: There are no coloring conflicts between wires on even/odd Metal-2 routing tracks.

In the 10-nm technology node used, the Metal-2 pitch was assumed to be $M2_{\text{pitch}} = 48$ nm, which corresponds to a 75% scaling from the 16-nm node [25]. Thus, the pitch between even/odd routing tracks was 96 nm ($2 \times M2_{\text{pitch}}$), which was larger than the resolution limit of 193-nm photolithography. Hence, Metal-2 wires on even/odd routing tracks are free of SADP coloring conflicts. Moreover, if the Metal-2 wires on the same or neighbor routing tracks were carefully designed to be SADP-friendly, then the line-space array decomposition method [10] from Fig. 1 could be deployed. To achieve the line-space array, each wire on the same routing track is extended and merged. As illustrated in Fig. 1(c), we place mandrel features on nonadjacent lines and add additional mandrels when necessary. In the spacer is dielectric process, the final layout patterns are defined as trim mask NOT spacer. Hence, we can design the trim mask efficiently as shown in Fig. 1(d). After choosing to adhere to line-space array decomposition, a second observation follows.

Observation 2: A single color is assigned to metal patterns on even routing tracks. The alternate color is then assigned to metal patterns on odd routing tracks.

Metal-2 routing is becoming increasingly congested as we continue to scale toward the 10-nm technology node because of the increasing density of transistors and SC I/O pins. Hence, increasing Metal-2 congestion leads to a higher likelihood of having Metal-2 wires on neighbor tracks at the same time, which leads to the layout patterns illustrated in Fig. 4. In Fig. 4(a), a solid edge denotes coloring conflict and a dashed edge denotes a potential coloring conflict. Fig. 4(b) demonstrates how a dashed edge changes to solid edge if we assign different colors to Metal-2 wires on the same track, which leads to an odd-cycle conflict in the coloring graph. SADP technique does not allow stitches during the layout decomposition stage, which means odd-cycle conflicts must be strictly forbidden in SADP-friendly layout patterns [7]. The color assignment strategy from Observation 2 helps to avoid potential odd cycles in the coloring stage for SADP-friendly layout.

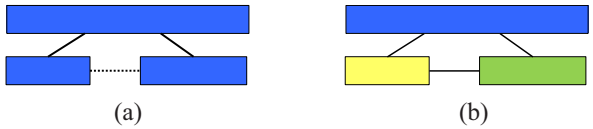

Fig. 4. Potential odd-cycle conflicts of the coloring graph. (a) Potential Metal-2 layout. (b) Potential odd-cycle conflicts.

TABLE I
SADP Related Notations

Metal-2 layer	MetalWidth	w
	MetalSpace	s
	minMetalArea	A_{min}
Spacer deposits	SpacerDepositWidth	w_{sp}
Trim mask	minResistWidth	w_r
	minResistSpace	s_r
	EtchBias	w_e
Design rules	minMetalLength ($Rule\ 0$)	l_0
	OnTrackSpace ($Rule\ 1$)	l_1
	OffTrackOverlap ($Rule\ 2$)	l_2
	OffTrackSpace ($Rule\ 3$)	l_3
	OffTrackOffset ($Rule\ 4$)	l_4

TABLE II
Design Rule Formulation

$Rule\ 0$	$l_0 \geq \frac{A_{min}}{w}$
$Rule\ 1$	$l_1 \geq w_r - 2 \cdot w_e$
$Rule\ 2$	$l_2 \geq s_r + 2 \cdot w_e$
$Rule\ 3$	$l_3 \geq \sqrt{(w_r - 2 \cdot w_e)^2 - {w_{sp}}^2}$
$Rule\ 4$	$l_4 \geq w_r \quad \text{or} \quad l_4 = 0$

B. SADP-Specific Design Rules

To enable layout design that is compatible with line-space array decomposition, we need to formulate SADP constraints into specific design rules. According to Observations 1 and 2, we define the four design rules shown in Fig. 5 that enforce SADP-friendly layout using 1-D relationships.

Table I defines SADP-specific notations for the 10-nm technology node [26]. First, the minimum area constraint of Metal-2 layout is converted to the minimum wire length design rule (l_0) due to the fixed width of Metal-2 wires. Then, we define the space between Metal-2 line ends on the same routing track as *OnTrackSpace*(l_1), as shown in Fig. 5(a). We use *OffTrackOverlap*(l_2) and *OffTrackSpace*(l_3) to define the prohibited region for the anti-parallel Metal-2 wires [26], as shown in Fig. 5(b) and (c), respectively. Finally, for parallel Metal-2 wires illustrated in Fig. 5(d), the line-end design constraint is defined as *OffTrackOffset*(l_4). Table II summarizes the design rules for Metal-2 layout patterns [17], [26].

C. Via-1 Design Rules

In this paper, the minimum coloring distance (d_0) and minimum different mask distance (d_1) are introduced to enable LELE friendly Via-1 assignment. Suppose, successful layout decomposition of Via-1 patterns from Fig. 6(a) and (b), d_0 denotes the minimum center to center spacing of Via-1s on the same mask and d_1 denotes the minimum center to center spacing of Via-1s on different masks.

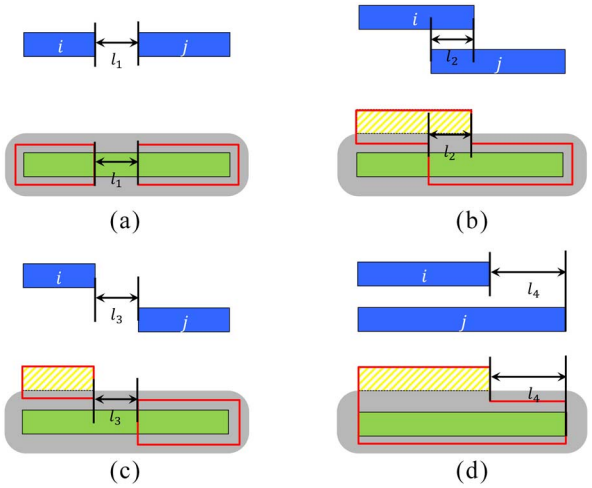

Fig. 5. SADP-specific design rule formulation. (a) *OnTrackSpace* $\geq l_1$. (b) *OffTrackOverlap* $\geq l_2$. (c) *OffTrackSpace* $\geq l_3$. (d) *OffTrackOffset* $\geq l_4$ or *OffTrackOffset* $= l_4 = 0$.

Fig. 6. (a) Via-1 patterns. (b) Layout decomposed to two masks and Via-1 design rules.

III. Problem Formulation

The SC level PAO involves both LELE-aware Via-1 assignment and SADP-aware Metal-2 wire design. Our I/O pin access design and optimization is based on practical layout of SCs. For each cell in the library, we observe that I/O pins generally exist on either the Metal-1 or Metal-2 layer. To properly formulate the pin access design and optimization problem, we have the following definitions.

Definition 1 (Hit Point): The overlap of a Metal-2 routing track (which is predetermined by the place and route tool) and an I/O pin shape is defined as a hit point for that particular I/O pin.

It can be observed that each hit point determines the range of positions for the corresponding Via-1 (the Metal-1 to Metal-2 connection). In the 10-nm technology used, Metal-2 is unidirectional and runs horizontally. Thus, for each hit point, there are two accessing directions possible, either from left to right or from right to left. To access one hit point, we first need to determine the legal location for Via-1 connection. Then, we need to design the Metal-2 wire assuming one accessing direction for the hit point. In order to connect to every I/O pin in a cell, we need to determine the accessing directions for a set of hit points. Hence, we have several definitions as follows.

Definition 2 (Hit Point Combination): A set of hit points (with a defined access direction, left or right) where each I/O pin in the SC is accessed exactly once is defined as a hit point combination for that cell.

Definition 3 (Valid Hit Point Combination): If a hit point combination induces zero design rule violations, it is considered a valid hit point combination. Otherwise, it is considered to be invalid.

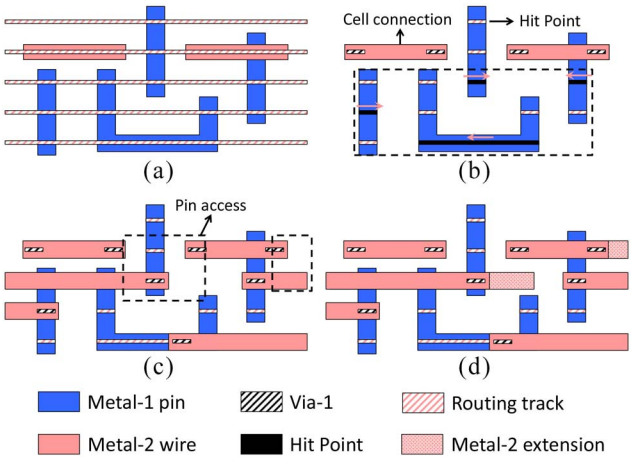

Fig. 7. Simple example for pin access design. (a) Metal-2 tracks and cell layout. (b) Hit points for each I/O pin and a hit point combination. (c) Metal-2 wires for pin access. (d) Line end extension.

Definition 4 (Valid Hit Point): If a hit point can be accessed from both directions within some valid hit point combinations for one cell, it is considered a valid hit point. Otherwise, it is considered to be invalid.

A simple example of the I/O pin access design for one hit point combination is shown in Fig. 7. Fig. 7(a) demonstrates the Metal-1 I/O pin layout and Metal-2 routing tracks running horizontally above Metal-1. Fig. 7(b) shows how the hit points, which represent valid Via-1 locations for I/O pin access, are derived from the overlap of Metal-2 routing tracks and the Metal-1 I/O pins. It can be observed that, for most hit points, the length of the hit point is short because it is decided by the minimum width of the vertical Metal-1. However, if a Metal-1 wire runs horizontally, this leads to a long hit point, which allows more flexibility for the Via-1 position. The set of hit points within the dashed box in Fig. 7(b) shows one hit point combination and the pink arrows denote the arbitrary accessing directions chosen for the hit points. Fig. 7(c) illustrates one way to access the cell using that hit point combination. After choosing one hit point for each I/O pin and the accessing direction for that hit point, we need to determine optimal Via-1 locations for that hit point combination such that the final Via-1 patterns are LELE friendly. Given LELE-legal Via-1 assignment, the Metal-2 wires can be designed for pin access, accounting for the minimum enclosure design rule for Metal-2 over Via-1. However, the dashed boxes in Fig. 7(c) denotes all pairs of line ends that cause hot spots in trim mask designs. Fig. 7(d) demonstrates that we can make use of line-end extension techniques to fix those hot spots in the trim mask.

In SC design, it is nontrivial to achieve an optimal LELE-aware Via-1 assignment or to determine whether all hot spots are fixable via line-end extension techniques. Furthermore, the engineering efforts and iterations involved to fix all of the potential hot spots across the SC library is too large for the average layout design team. Therefore, a general methodology is needed to determine optimal Via-1

assignment and design the Metal-2 wires for pin access and within-cell connections simultaneously. We can now define the SADP-aware PAO problem as follows.

Problem 1 (PAO): Given the SC layout and a specific hit point combination, determine whether or not it is possible to achieve an optimal LELE-friendly Via-1 assignment and subsequently optimize the Metal-2 wires for pin access and within-cell connections given the Via-1 assignment under SADP constraints. If possible, show legal Via-1 assignment and all SADP-friendly Metal-2 wires.

Moreover, as shown in Fig. 7(b), we may have multiple hit points for one I/O pin, which leads to numerous hit point combinations for one cell. For one SC, we define the PICO problem as follows.

Problem 2 (PICO): Given the SC layout, the PICO problem is to show all Via-1 assignment and Metal-2 wiring cases with successful PAOs and maximize the pin access flexibility under LELE and SADP constraints.

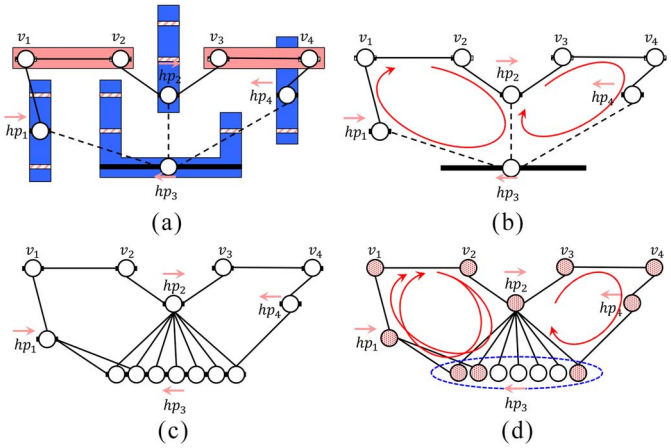

Fig. 8. Optimal Via-1 assignment for one hit point combination. (a) Via-1s for cell connections and hit points for pin access. (b) Initial conflict graph. (c) Grid-based segmentation of the long hit point. (d) Odd cycles in conflict graph.

IV. PAO

Given a hit point combination, we seek the optimal double patterning friendly Via-1 assignment. Based on the legal Via-1 assignment, we predesign the Metal-2 wires for pin access. Then, we propose an MILP-based method to enable efficient SADP-aware Metal-2 design. Finally, the linear programming (LP) relaxation is presented to explore the trade-off between run time and performance for the PAO.

A. LELE-Aware Via-1 Assignment

As the first step, we need to determine the Via-1 location for each hit point within the given hit point combination such that the Via-1s are double patterning friendly.

In general, there are two kinds of Via-1s in the SC level. The first kind is for internal Metal-1 to Metal-2 (within-cell) connections that are needed because of the complexity of SC layout in advanced technology nodes. These connections can not be modified in the PAO stage. The other kind is for the pin access connections from Metal-2 wires to Metal-1 I/O pins, which is the design target in the Via-1 assignment problem. For instance, in Fig. 8(a), there are four internal Via-1s denoted as $v_1 - v_4$ and our target is to determine the location of Via-1s for the four hit points denoted as $\text{hp}_1 - \text{hp}_4$. Specifically, we can easily drop the Via-1s in the center of short hit points, such as hp_1, hp_2, and hp_4. However, long hit points, such as hp_3, allow for more flexibility for the Via-1 location. We can build the conflict graph for the Via-1 layer given a specific hit point combination, as illustrated in Fig. 8(b). We put a solid edge between conflicting Via-1s of which locations are decided. A dashed edge is added when the location of some Via-1 is not decided and there exists potential conflict between two Via-1s. For example, the Via-1 associated with the long hit point in Fig. 8(b) always conflicts with the Via-1 for hp_2, but only conflicts with that for hp_1 when dropped on the left end and conflicts with that for hp_4 when dropped on the right end.

To make the best use of the flexibility from the long hit point, we propose the grid based segmentation, which leads to multiple potential Via-1 candidates for a hit point. For example, we can see several Via-1 candidates for hp_3 in Fig. 8(c). The grid size depends on the granularity needed. To properly define the problem, related notations are given in Table III. In addition, there is an accessing direction assigned to each hit point within the given hit point combination. Hence, it can be clearly observed that we prefer the legal Via-1 location to be on the right end of hp_3 because the hit point will be accessed from right to left. The right most legal Via-1 candidate for hp_3 leads to shortest Metal-2 wire to access that hit point. To differentiate multiple Via-1 candidates for the long hit point, we assign cost to each candidate according to the accessing direction and distance from the Via-1 location to the left/right end of the hit point. For instance, the cost of the right most Via-1 for the long hit point in Fig. 8(c) will be 0 and the cost of second right most Via-1 will be g_0. In general, from right to left, the cost of the nth Via-1 will be $(n-1) * g_0$. We assign zero cost to internal Via-1s and Via-1s associated with short hit points. After the grid based segmentation and cost assignment for each Via-1 candidate, we can build the whole conflict graph for the Via-1 layer, as illustrated in Fig. 8(d). There is no conflict among Via-1 candidates for the same hit point.

TABLE III
LELE-AWARE VIA-1 ASSIGNMENT NOTATIONS

v_k	the k_{th} Via-1 for cell connections
hp_k	the k_{th} hit point within given hit point combination
hp_k^m	the m_{th} candidate Via-1 location for k_{th} hit point
S_v	the LELE-Aware Via-1 assignment
S_v^o	the optimal LELE-Aware Via-1 Assignment
g_0	the Via-1 grid size

The overall objective of this step is to choose a set of Via-1 locations, including Via-1s for internal cell connection and one Via-1 for each hit point, such that the total cost is minimized and the Via-1 layer is double patterning friendly. It is a well established argument that the layout is double patterning friendly if and only if the conflict graph deduced does

Algorithm 1 Via-1 Assignment Algorithm

Require: a set of Via-1s for cell connections (S_c), a hit point combination (*HPC*) and Via-1 grid size (g_0);
1: Define *CFGraph* as the conflict graph for Via-1 layer;
2: Define S_{via} as the 2-D set of Via-1s;
3: **for** each element *hp* in $HPC \cup S_c$ **do**;
4: Define S_{pos} as the set of Via-1s for *hp*;
5: **if** *hp* is a long hit point **then**;
6: Segment *hp* in the grid size g_0;
7: Push the center of each segment to S_{pos};
8: **else**
9: Push the center of *hp* to S_{pos};
10: **end if**
11: **for** each via location *pos* in S_{pos} **do**;
12: Add *pos* to *CFGraph* and assign cost;
13: Detect and add conflict edges;
14: **end for**
15: Push S_{pos} to S_{via}
16: **end for**
17: Define *oddCycles* as the 2-D set of nodes in *CFGraph*;
18: Report all odd cycles in *CFGraph* to *oddCycles*;
19: Define S_v^o as the optimal Via-1 assignment;
20: S_v^o = DFS(S_{via}, *oddCycles*);
21: **if** $S_v^o = \emptyset$ **then**;
22: *HPC* is invalid;
23: **else**
24: S_v^o is the optimal legal Via-1 assignment;
25: **end if**

Algorithm 2 DFS Algorithm

Require: a 2-D set of Via-1s (S_{via}) and a set of forbidden cycles (*oddCycles*);
1: **function** DFS(S_{via}, *oddCycles*)
2: Construct search tree based on S_{via};
3: Define list S_v for traversed nodes;
4: Define $S_v^o = \emptyset$ as the optimal Via-1 assignment;
5: Define *minCost* as the cost for S_v^o;
6: *minCost* = *HUGE*;
7: **repeat**
8: $S_v = \emptyset$;
9: **while** traverse from root to leaf **do**;
10: **if** one element of *oddCycles* is the sub-set of S_v or different mask rule check failed **then**
11: Go to next node on the same level;
12: **else**
13: Push(*currentnode*, S_v);
14: **end if**
15: **end while**
16: **if** cost of $S_v <$ *minCost* **then**
17: *minCost* = cost of S_v;
18: $S_v^o = S_v$;
19: **end if**
20: **until** all paths exhausted
21: return S_v^o;
22: **end function**

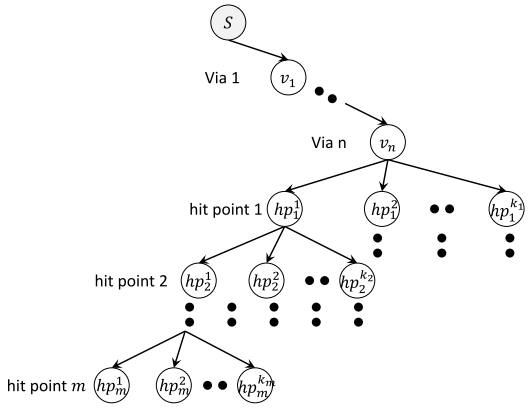

Fig. 9. Search tree for the DFS method.

not contain odd cycles [20]. This means that odd cycles are forbidden in the optimal legal Via-1 assignment.

Algorithm 1 demonstrates the details for achieving optimal LELE-friendly Via-1 assignment. Lines 3–16 explain how to build the conflict graph given Via-1s for cell connection and a specific hit point combination. In line 18, all of the odd cycles in the conflict graph are reported [e.g., the three odd cycles in Fig. 8(d) would be reported]. If an odd cycle is detected in the conflict graph, the conflicting set of Via-1 candidates cannot be chosen simultaneously during legal Via-1 assignment. Kahng *et al*. [20] deployed the breadth first search and double linked list to iteratively report odd cycles. In this case, our target is different because we need to find all of the odd cycles simultaneously in a given conflict graph. Hence, the DFS technique is used. We keep track of the DFS stack while coloring the conflict graph. Once a coloring conflict is reported for double patterning, we backtrack the DFS stack and report the cycle associated with current coloring conflict, which is repeated until all nodes are visited. The optimal Via-1 assignment is achieved in line 20 and the details are given in Algorithm 2. The optimal Via-1 assignment is an empty set if no legal Via-1 assignment can be found, as explained in lines 21–25. Assume n hit points for the input hit point combination and the number of valid Via-1 candidates for the ith hit point is $k_i, 0 \leq i \leq n-1$, the total number of Via-1 combinations enumerated will be $\Pi_{i=0}^{i=n-1} k_i$. Due to the linear time complexity of DFS, the theoretical run time bound for Algorithm 1 will be $O(k^n), k = \max_i k_i$. The time complexity for Algorithm 1 grows exponentially with the number of I/O pins, but most SC libraries have cell I/O pin counts bounded to the order of 10. This means that for a typical SC library, Algorithm 1 can execute in a reasonable runtime.

Algorithm 2 illustrates the DFS-based technique to achieve the legal set of Via-1 candidates while minimizing cost. In line 2, we construct a search tree out of all Via-1 candidates. Fig. 9 demonstrates the construction of the search tree from Via-1s ($v_1 - v_n$) for cell connections and a hit point combination ($\text{hp}_1 - \text{hp}_m$) given the union set, S_{via}. The root, S, will be a virtual node for the tree. A path from root to leaf gives a potential Via-1 assignment for the cell. The ith, ($1 \leq i \leq n$) level of the search tree contains Via-1s for the cell connections.

The *j*th, $(n+1 \leq j \leq m+n)$ level of the search tree contains Via-1 candidates for the corresponding hit point. Specifically, only one Via-1 candidate will be included for a short hit point, but various candidates may be added for a long hit point, depending on the selected grid size. While traversing from root to node, before adding a new node to the path, we ensure the current set of Via-1 candidates (S_v) does not contain odd cycles. In particular, the different mask rule for the Via-1 layer defines the minimum center to center spacing due to the overlay constraint for double patterning, which should be strictly satisfied in Via-1 design. We continue the traversal if the node is legal on the path. Otherwise, we go to the next node on the same level, which prunes the redundant Via-1 assignment relevant to that node. Lines 16–19 show that once a legal Via-1 assignment is found, only the solution with minimized cost will be stored. The optimal Via-1 assignment can be achieved when all paths are exhausted in line 20.

The double patterning aware Via-1 assignment yields a legal Via-1 location for each hit point within the hit point combination. Furthermore, the solution of Algorithm 1 decides the boundary condition for the SADP-aware pin access design of the Metal-2 layer.

B. Pin Access Predesign

Given optimal Via-1 assignment and the accessing direction for the corresponding hit point, we can determine the line end position of the Metal-2 wire for pin access accounting for the minimum enclosure design rule for Metal-2 over Via-1. For pin access design, we focus on SADP-aware layout optimization within a SC boundary. Hence, if one hit point is next to the right boundary of a cell and the access direction is from the right, the right line end of the corresponding Metal-2 wire will be extended to the right boundary. We have similar predesign if the hit point is accessed from the left boundary of a cell. Fig. 7(c) is an example of Metal-2 wires for pin access after the predesign stage. This is the most common hit point access scenario from the SC perspective. Our primary goal is to achieve the first-order pin accessibility figure-of-merit. Hence, we exclude special scenarios like Metal-3 wire going down directly through two vias (Metal-3 to Metal-2 and Metal-2 to Metal-1), and also pin access through complex maze routing, instead of straight routing. The predesign method induces the following observation.

Observation 3: For Metal-2 wires after the predesign stage, right line ends can only be extended to the right and left line ends can only be extended to the left.

C. SADP-Aware Pin Access

As illustrated in Fig. 7(c), if we simply use the Via-1 locations to determine the line end of Metal-2 wires, the SADP constraints may invalidate some hit point combinations. The line-end extension techniques enable us to legalize the Metal-2 layout and ensure SADP-friendly design. The conventional layout migration issue has been formulated as a LP problem in [27]. A similar approach has also been deployed to deal with LELE double patterning layout decomposition in [28]. In addition, the line-end extension techniques have also been deployed in [18] and [19] to improve the manufacturability of the cut mask process instead of the trim mask process in this paper. Zhang *et al*. [18] proposed a greedy optimal algorithm to minimize the effective gaps in the cut mask process. The constrained shortest path algorithm is presented to optimize the cut mask cost. Instead of mask cost reduction, this paper is design rule oriented and guarantees to find an optimal solution with minimum amount of line-end extensions if a feasible solution exists for the given Via-1 assignment from Section IV-A. The greedy algorithm in [18] aims at the gap distribution optimization in cut masks, which is different from the trim mask optimization in this paper. The techniques used in [19], [27], and [28] cannot be directly applied to SADP-aware I/O pin access design because the relative order of the metal line ends may change during the line end extension stage, as shown in Fig. 2(a). Instead, we propose an MILP-based optimization methodology to determine the Metal-2 wire design for each specific hit point combination. Table IV shows the notations for the variables used in our formulation. We will first give the mathematical formulation for our SADP-aware pin access problem. Then, we transfer the mathematical formulation to an MILP formulation. The results of the MILP can determine whether feasible solutions exist for the Metal-2 line ends of a particular hit point combination. If feasible solutions exist, the line end positions of each Metal-2 wire are decided while minimizing the total amount of line end extension.

TABLE IV
SADP-AWARE PIN ACCESS NOTATIONS

c_L, c_R	left or right boundary of cell
c_W	cell width, $c_W = c_R - c_L$
S_m	set of Metal-2 wires
n	total number of Metal-2 wires
S_k	set of pairs of wires for rule $k, \forall k \in \{1,2,3,4\}$
x_{iL}, x_{iR}	the left or right line end of i_{th} wire
x^0_{iL}, x^0_{iR}	the initial line ends of i_{th} wire

1) Mathematical Formulation: Observation 3 allows us to quantify the total amount of extension in terms of line-end positions. It is known that line end extension techniques benefit SADP-based wires [17]. However, in next generation technology nodes, the routing resources are becoming increasingly limited, so line-end extensions of Metal-2 wires should be used judiciously. Additionally, line end extensions can potentially increase both coupling capacitance and ground capacitance on Metal-2 routes. Therefore, line-end extension minimization is a necessity for PAO. The minimization of the total amount of line-end extensions is formulated as the objective function, as shown in (1).

Constraints (1a)–(1c) define the line-end extension limits and minimum wire length design rule (Rule 0 in Table II) for each Metal-2 wire. The initial relative order can be determined for each pair of Metal-2 wires. Suppose the *i*th wire is on the left of the *j*th wire, as demonstrated in Fig. 5(a). Constraint (1d) is formulated to define Rule 1. In set S_2, the line ends originally overlap each other and constraints (1e) and (1f) interpret Rule 2. In set S_3, the line

ends initially have no overlap. After extension, the line ends may or may not overlap each other. Constraint (1g) satisfies Rules 2 or 3. Then, constraints (1h) and (1i) are formulated to specify Rule 4 for each pair of Metal-2 wires in set S_4

$$\min \sum_{i=0}^{n-1} \left(x_{iL}^0 - x_{iL}\right) + \left(x_{iR} - x_{iR}^0\right) \tag{1}$$

$$\begin{aligned}
\text{s.t.}\quad & c_L \le x_{iL} \le x_{iL}^0 && \forall i \in S_m && \text{(1a)}\\
& x_{iR}^0 \le x_{iR} \le c_R && \forall i \in S_m && \text{(1b)}\\
& x_{iR} - x_{iL} \ge l_0 && \forall i \in S_m && \text{(1c)}\\
& x_{jL} - x_{iR} \ge l_1 && \forall (i,j) \in S_1 && \text{(1d)}\\
& x_{iR} - x_{jL} \ge l_2 && \forall (i,j) \in S_2 && \text{(1e)}\\
& x_{jR} - x_{iL} \ge l_2 && \forall (i,j) \in S_2 && \text{(1f)}\\
& x_{jL} - x_{iR} \ge l_3 \text{ or } x_{iR} - x_{jL} \ge l_2 && \forall (i,j) \in S_3 && \text{(1g)}\\
& |x_{iL} - x_{jL}| \ge l_4 \text{ or } x_{iL} - x_{jL} = 0 && \forall (i,j) \in S_4 && \text{(1h)}\\
& |x_{iR} - x_{jR}| \ge l_4 \text{ or } x_{iR} - x_{jR} = 0 && \forall (i,j) \in S_4. && \text{(1i)}
\end{aligned}$$

2) MILP Formulation: Here, we show how to convert (1) into an MILP formulation. We can simplify the objective function by omitting item x_{iL}^0 and x_{iR}^0, which are constants for a specific hit point combination. In addition, we also need to convert constraints (1g)–(1i) to linear constraints based on the big-M transformation [29].

Note that $|x_{iL} - x_{jR}| \le c_W$, $\forall i, j \in S_m$ and c_W is the width of the cell. Hence, in the SC level, the cell width c_W is an appropriate big-M parameter for our formulation. Constraint (1g) can be formulated as linear constraints (2a)–(2c) given below. s_k is an additional integer variable introduced so that both constraints can be satisfied at the same time

$$\begin{aligned}
& x_{jL} - x_{iR} + (c_W + l_3) \cdot s_k \ge l_3 && \text{(2a)}\\
& x_{iR} - x_{jL} + (c_W + l_2) \cdot (1 - s_k) \ge l_2 && \text{(2b)}\\
& s_k \in \{0, 1\} \qquad \forall (i,j) \in S_3. && \text{(2c)}
\end{aligned}$$

Similarly, constraints (1h) and (1i) can also be converted to linear constraints by introducing integer variables as follows:

$$\begin{aligned}
& x_{jL} - x_{iL} + (c_W + l_4) \cdot s_{m1} \ge l_4 \cdot (1 - t_{n1}) && \text{(2d)}\\
& x_{iL} - x_{jL} + (c_W + l_4) \cdot (1 - s_{m1}) && \text{(2e)}\\
& \quad \ge l_4 \cdot (1 - t_{n1}) + (c_W + l_4) \cdot t_{n1} && \text{(2f)}\\
& s_{m1} + t_{n1} \le 1, s_{m1}, t_{n1} \in \{0, 1\} \quad \forall (i,j) \in S_4 && \text{(2g)}\\
& x_{jR} - x_{iR} + (c_W + l_4) \cdot s_{m2} \ge l_4 \cdot (1 - t_{n2}) && \text{(2h)}\\
& x_{iR} - x_{jR} + (c_W + l_4) \cdot (1 - s_{m2}) && \text{(2i)}\\
& \quad \ge l_4 \cdot (1 - t_{n2}) + (c_W + l_4) \cdot t_{n2} && \text{(2j)}\\
& s_{m2} + t_{n2} \le 1, s_{m2}, t_{n2} \in \{0, 1\} \quad \forall (i,j) \in S_4. && \text{(2k)}
\end{aligned}$$

To summarize, the MILP formulation is shown in (2). The optimization results will decide whether it is possible to achieve a legal solution for the Metal-2 design given the optimal Via-1 assignment of one hit point combination. In particular, the feasible solution of the MILP formulation consists of the legal line-end position of each Metal-2 wire with the minimum amount of extension

$$\begin{aligned}
& \min \sum_{i=0}^{n-1} (x_{iR} - x_{iL}) && \text{(2)}\\
& \text{s.t.}\quad \text{(1a)} - \text{(1f)}\\
& \qquad \text{(2a)} - \text{(2h)}.
\end{aligned}$$

3) LP Relaxation: In the MILP formulation (2), integer variables are introduced to allow changing the relative order of metal line ends. This guarantees to determine the feasibility of the given Via-1 assignment with minimized total amount of line end extensions. However, the MILP formulation may lead to huge run time for large problem scale. To explore the necessity of the MILP formulation and the possibility of speed-up, we relax the MILP formulation (2) into the LP formulation. Specifically, the relative order of line ends are predetermined based on the initial position of metal lines, which helps to convert constraints (1g)–(1i) to linear constraints. For example, we can split the constraint (1g) into linear constraints (2l) and (2m) according to the initial metal line-end positions. Similarly, the constraints (1h) and (1i) can be converted to constraints (2n)–(2p) and constraints (2q)–(2s), respectively

$$\begin{aligned}
& x_{jL} - x_{iR} \ge l_3 \quad \text{if } x_{jL}^0 - x_{iR}^0 \ge l_3 && \forall (i,j) \in S_3 && \text{(2l)}\\
& x_{iR} - x_{jL} \ge l_2 \quad \text{if } x_{jL}^0 - x_{iR}^0 < l_3 && \forall (i,j) \in S_3 && \text{(2m)}\\
& x_{iL} - x_{jL} \ge l_4 \quad \text{if } x_{iL}^0 - x_{jL}^0 \ge l_4 && \forall (i,j) \in S_4 && \text{(2n)}\\
& x_{jL} - x_{iL} \ge l_4 \quad \text{if } x_{jL}^0 - x_{iL}^0 \ge l_4 && \forall (i,j) \in S_4 && \text{(2o)}\\
& x_{iL} = x_{jL} \quad \text{if } |x_{iL}^0 - x_{jL}^0| < l_4 && \forall (i,j) \in S_4 && \text{(2p)}\\
& x_{iR} - x_{jR} \ge l_4 \quad \text{if } x_{iR}^0 - x_{jR}^0 \ge l_4 && \forall (i,j) \in S_4 && \text{(2q)}\\
& x_{jR} - x_{iR} \ge l_4 \quad \text{if } x_{jR}^0 - x_{iR}^0 \ge l_4 && \forall (i,j) \in S_4 && \text{(2r)}\\
& x_{iR} = x_{jR} \quad \text{if } |x_{iR}^0 - x_{jR}^0| < l_4 && \forall (i,j) \in S_4. && \text{(2s)}
\end{aligned}$$

Therefore, the LP relaxation formulation is illustrated in (3). As we can see in Section VI, this formulation can not optimally determine the feasibility of the given Via-1 assignment but can significantly reduce the run time, compared with the MILP formulation in (2)

$$\begin{aligned}
& \min \sum_{i=0}^{n-1} (x_{iR} - x_{iL}) && \text{(3)}\\
& \text{s.t.}\quad \text{(1a)} - \text{(1f)}\\
& \qquad \text{(2l)} - \text{(2s)}.
\end{aligned}$$

V. PICO

Previously, we have shown that the double patterning aware Via-1 assignment and MILP-based optimization for Metal-2 wires determine whether a single hit point combination is valid or not. If it is valid, we can achieve optimal LELE friendly Via-1 assignment and subsequently optimize the Metal-2 wires for pin access and cell connection simultaneously given the Via-1 assignment. However, as shown in Fig. 7, multiple hit points for one I/O pin lead to numerous hit point combinations for one SC. In general, the more valid hit point combinations we have for one cell, the more flexibility we can provide to

Algorithm 3 PICO Algorithm

Require: Cell layout and Metal-2 routing tracks;
1: Define C as set of Metal-2 wires for within-cell connections;
2: Define IO as set of I/O pins;
3: Define S_{hp} as the 2-D set of hit points for each I/O pin;
4: **for** each pin $p_k \in IO$ **do**
5: Get the set of hit points, P_k, for pin p_k;
6: Add P_k to S_{hp};
7: **end for**
8: Define *MTable* as table of Metal-2 layout design;
9: Define *HPC* as an empty set for hit point combination;
10: *MTable* = Backtracking(*HPC*, S_{hp}, C);
11: **for** each entry H_k in *MTable* **do**;
12: **if** PAO for $H_k \cup C$ is feasible **then**
13: Replace H_k with the solution from PAO;
14: **else**
15: Delete H_k;
16: **end if**
17: **end for**

Algorithm 4 Backtracking Algorithm

Require: a hit point combination (*HPC*), a 2-D set of hit points for IO pins (S_{hp}) and a set of Metal-2 wires (C);
1: **function** BACKTRACKING(*HPC*, S_{hp}, C)
2: Define *pinCnt* as size of S_{hp};
3: Define *hpCnt* as size of *HPC*;
4: **if** *hpCnt* = *pinCnt* **then**;
5: Add *HPC* to *Table*;
6: **else**
7: **for** each hit point (*hp*) in the $hpCnt_{th}$ row of S_{hp} **do**;
8: **if** Check(*hp*, *HPC*, *C*)=True **then** ▷ The check heuristics for Backtracking
9: Push(*hp*, *HPC*);
10: Backtracking(*HPC*, S_{hp}, C);
11: **end if**
12: **end for**
13: **end if**
14: Pop the last element of *HPC*;
15: **end function**

the routing stage. Thus, we extend the PAO to validate all hit point combinations of a SC.

The overall algorithm for the PICO is given in Algorithm 3. First, as shown in lines 1–7, the preprocessing steps determine the set of hit points for each I/O pin. Then, in line 10, the backtracking method is proposed to obtain a table of potential valid hit point combinations for the SC. From lines 11–17, we call the double patterning aware Via-1 assignment and MILP-based optimization (2) for each entry in the table of potential hit point combinations. Assume n I/O pins for the input SC and h_i is the number of hit points for the ith I/O pin, $0 \leq i \leq n-1$, the number of hit point combinations enumerated is $\Pi_{i=0}^{i=n-1} 2 * h_i$. The goal is to optimize each hit point combination, individually, hence, the time complexity will be $O((2 * h)^n)$, $h = \max_i h_i$. The parametric constant associated with the run time complexity is given by the PAO optimization.

For each cell, all valid pin access designs are stored in a table, which can be incorporated into the SC library design. Hence, we have maximized the pin access flexibility of one cell for the routing stage.

Algorithm 4 illustrates the backtracking method that yields a table of potential valid hit point combinations. Lines 4 and 5 show that a potential valid hit point combination can be extracted when the backtracking stack (HPC) is full. Lines 7–13 illustrate the steps to explore the solution space for the hit point combination enumeration. In particular, we check the compatibility of the newly added hit point with those in the stack and the Metal-2 wires for within-cell connections. The backtracking step is given in line 14.

The following check heuristics help to prune out invalid hit point combinations.

1) Avoid two hit points that are close to each other and on the same track.
2) Existing Metal-2 wires used for within-cell connection invalidate the hit points they cover as well as hit points that are too close in proximity.

Here, the "close" means that the two hit points are on the same routing track and the tip-to-tip spacing is less that a certain threshold, which is l_1 in Table I. It should be noted that we can further consider other pruning metrics during the backtracking stage. For example, cell robustness metrics, such as pin density, are closely related to the pin access design at the SC level and it could be another metric used to prune out invalid hit point combinations.

VI. Experimental Results

A. Experimental Setup

We have implemented our algorithm in C++ and tested it using an industrial 14-nm SC library that has been scaled and compacted to 10-nm representative dimensions. We use CBC [30] as our MILP solver and all experiments are performed on a Linux machine with a 3.33 GHz Intel Xeon CPU X5680. The width and space of Metal-2 wires are assumed to be 24 nm. The spacer deposit width is set as 24 nm. For trim mask design, the minimum resist width and space are set as 44 and 46 nm, respectively. The etch bias is set as 6 nm [26]. The industrial 10-nm design rules are set for the Via-1 layer and grid size g_0 for the segmentation of hit points is set as 2 nm. Next, we demonstrate the strength of our optimization methodology by showing the results from pin access design for specific hit point combinations, SCs and the entire SC library consisting of around 700 cells.

B. PAO Results

Fig. 10 demonstrates a typical cell layout design in the library we used. The I/O pins for this cell are on the Metal-1 layer. Due to the complexity of this cell, Metal-2 wires are used for within-cell connections. Fig. 10(a) shows the Metal-2 layout design if we simply use Via-1 locations after LELE aware Via-1 assignment to determine the line-end positions. A design rule check will reveal multiple violations in the

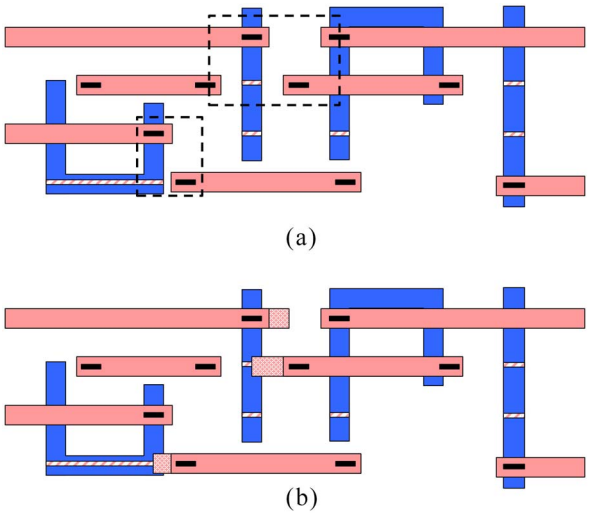

Fig. 10. PICO for one hit point combination. (a) Design rule violations in layout. (b) MILP-based optimization result.

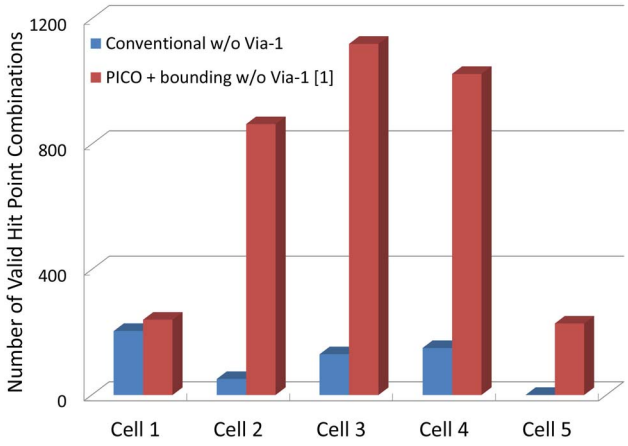

Fig. 11. Increase in the number of valid hit point combinations, without Via-1 rules.

dashed boxes. However, as illustrated in Fig. 10(b), the same Metal-2 wires for pin access and within-cell connections can be co-optimized to enable SADP-friendly layout. The MILP-based optimization ensures the minimum amount of line-end extension and avoids potential engineering efforts from design rule violation fixes.

C. PICO Results

The PICO problem motivates us to extend the optimization technique to each hit point combination of the SC based on the backtracking method. The pin access flexibility of each cell can be evaluated in terms of the number of valid hit point combinations or valid hit points for each I/O pin. As shown in Fig. 10, we may have various hit points for each I/O pin of the cell. We assume that Via-1 will be dropped in the center of the hit point and Via-1 locations determine the line-end positions of Metal-2 wires in conventional pin access design. The design rule checks for conventional design have been implemented as the baseline.

The conference version of [1] focused on the Metal-2 design and optimization without Via-1 rules. Fig. 11 shows

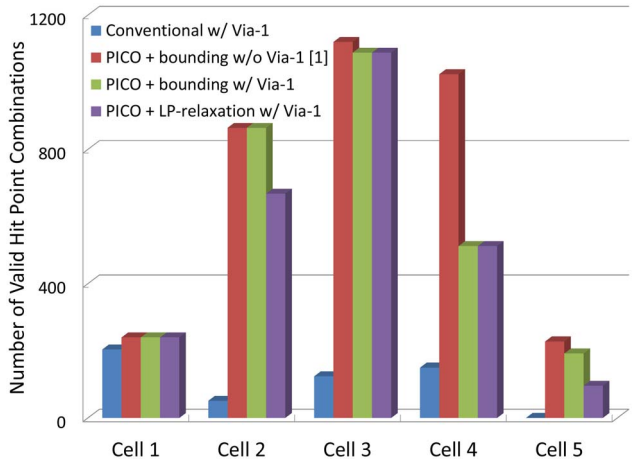

Fig. 12. Increase in the number of valid hit point combinations, with Via-1 rules.

the effectiveness of the PICO without Via-1 constraints for five typical cells in the 10-nm SC library. We observe that the improvement of the number of valid hit point combinations is cell dependent. For cell 1, the space between I/O pins is relatively large, which means pin access design can easily satisfy the LELE and SADP constraints and there is no significant improvement in terms of valid hit point combinations. However, we achieve a significant increase in the number of valid hit point combinations for other cells in Figs. 11 and 12. Particularly, conventional design rule checking invalidates all of the hit point combinations for cell 5, the layout of which is shown in Fig. 10. This invalidation is caused by the existing, within-cell connections in Metal-2, which illustrates why we need to simultaneously optimize the pin access wires along with the wires that already exist in a given cell's layout. Our optimization framework recovers nearly half of the total hit point combinations for cell 5. In this paper, we further consider the double patterning aware Via-1 assignment for pin access and within-cell connections. Fig. 12 demonstrates the impact on the number of valid hit point combinations when considering Via-1 rules. Here, we add an additional column as "PICO + LP-relaxation w/Via-1." This means the LP-relaxation is deployed in PICO instead of the MILP formulation used in "PICO + bounding w/Via-1." For the first three cells, the Via-1 rules have little impact on the results. For these cells, Via-1 rules are noncritical for the legal pin access design because the Via-1s in the final design are sparse. The MILP-based PAO still achieves a significant increase in the number of valid hit point combinations from conventional to PICO, even with Via-1 rules considered. For cell 4, despite the increase in the number of valid hit point combinations from conventional with Via-1 to PICO with Via-1, the amount of increase is much smaller than that in Fig. 11. Therefore, the double patterning aware Via-1 assignment is critical for the robust pin access design of this cell. In addition, with the LP relaxation speed-up in the fourth series of Fig. 12, all feasible hit point combinations are achieved for cells 1, 3, and 4. However, some valid hit point combinations are lost for cells 2 and 5 because the LP relaxation forbids the relative order change of the line-end positions.

TABLE V
INCREASE IN THE NUMBER OF VALID HIT POINTS FOR EACH I/O PIN

	Conventional w/o Via-1				Conventional w/ Via-1				PICO w/o Via-1				PICO w/ Via-1				PICO + LP-relaxation w/ Via-1			
pin index	#1	#2	#3	#4	#1	#2	#3	#4	#1	#2	#3	#4	#1	#2	#3	#4	#1	#2	#3	#4
Cell 1	2	3	5	-	2	3	5	-	2	3	5	-	2	3	5	-	2	3	5	-
Cell 2	0	0	3	2	0	0	3	2	2	2	4	7	2	2	4	7	2	2	4	7
Cell 3	3	3	3	6	3	3	3	6	3	3	3	6	3	3	3	6	3	3	3	6
Cell 4	1	0	1	7	1	0	1	7	2	2	2	8	1	2	2	8	1	2	2	8
Cell 5	0	0	0	0	0	0	0	0	1	3	2	3	1	2	2	3	1	1	2	3

For both conventional design and PICO optimization, we further evaluate the number of valid hit points based on the definition of the valid hit point shown in Section III. Compared to the conventional approach, PICO achieves an increase of valid hit points for each I/O pin of the cells, as demonstrated in Table V. We incorporate the results without Via-1 rules, i.e., "conventional w/o Via-1" and "PICO w/o Via-1" for comparison. We put a "-" in the entry of the table if the *i*th I/O pin does not exist for that particular cell. For instance, cell 1 only has three I/O pins, so pin #4 has a - in their entries since they do not apply. It is interesting to note that some cells like "cell 3" show zero improvement in the number of valid hit points in Table V, but show significant increases in the number of valid hit point combinations in Fig. 11. This example highlights the difference between hit points and hit point combinations, and the importance of enumerating and optimizing the combinations, not just the hit points themselves. This is due to the fact that hit points in a combination influence each other and can cause SADP violations in the combination. In isolation, a hit point may appear valid, but upon grouping into a combination, it may negatively impact other hit points. In addition, we see no decrease in valid hit points from conventional without Via-1 to conventional with Via-1, because the SADP specific design rule check qualitatively ensures the sparseness of Via-1s, which means legal Metal-2 wires tend to achieve legal Via-1 assignment. However, we see the decrease of valid hit points for the second I/O pin of cell 5 from PICO w/o Via-1 to "PICO w/Via-1." This implies that the LELE-friendly Via-1 assignment must be guaranteed before the MILP-based SADP-aware pin access design. Moreover, the results from the PICO with LP relaxation, denoted as PICO + LP-relaxation w/Via 1, are shown on last four columns. We can see that for Cell 5, the number of valid hit points for the second pin has decreased from "2" in PICO w/Via-1 to "1" in PICO + LP-relaxation w/Via-1. This implies that MILP formulation is necessary to achieve maximized pin access flexibility for the SC level optimization.

To gauge the library-wide effectiveness of our optimization framework, we also applied the proposed technique to each cell in our 10-nm library. For the sake of the correctness and completeness of the optimization framework, we only demonstrate the results from PICO with Via-1 rules. We calculated the ratio of valid hit point combinations of PICO over the conventional approach for each cell. In Figs. 13 and 14, "PICO + bounding" means the PAO with MILP formulation and check heuristics applied as discussed in Section V and "PICO + LP-relaxation" means PAO with LP relaxation applied. The histogram in Fig. 13 demonstrates the valid hit point combination ratio and the effectiveness of the PICO technique. We obtain 10× or more improvement for most cells and some cells achieve up to a 10 000× increase in the number of valid hit point combinations. This means that PICO has significantly improved the validation of the SC pin access design. We also evaluate the increase in the number of valid hit points for each I/O pin. The increase in percentage over the total number of hit points is calculated for each cell in the library and shown in Fig. 14. In our 10-nm experiments, we find that over 1/3 of the cells have 30% improvement in the number of valid hit points. From Figs. 13 and 14, we can see that, with the PICO + LP-relaxation, more cells tend to achieve less amount of increase in valid hit point and valid hit point combinations compared with the PICO + bounding.

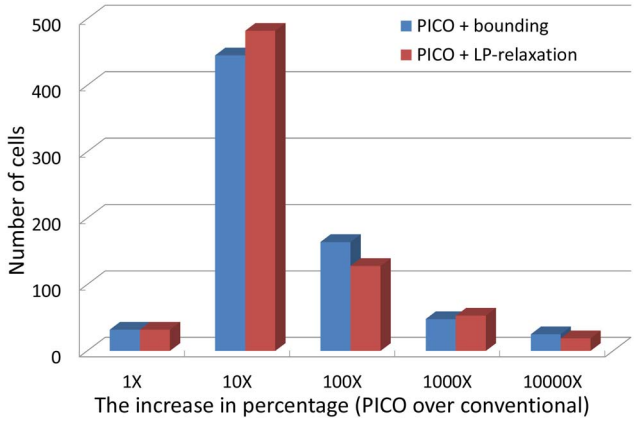

Fig. 13. Increase in ratio on the number of valid hit point combinations across the entire cell library.

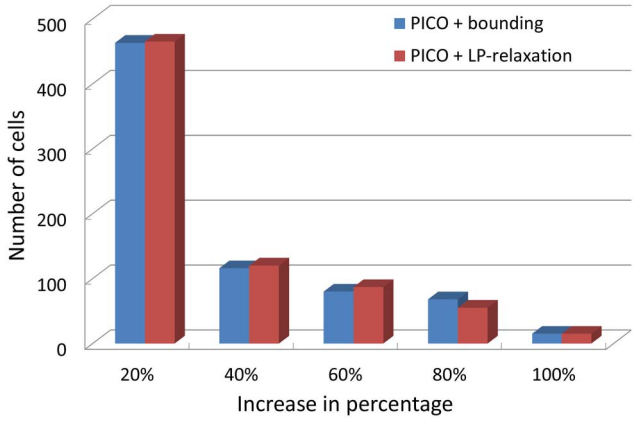

Fig. 14. Increase in percentage on the number of valid hit points across the entire cell library.

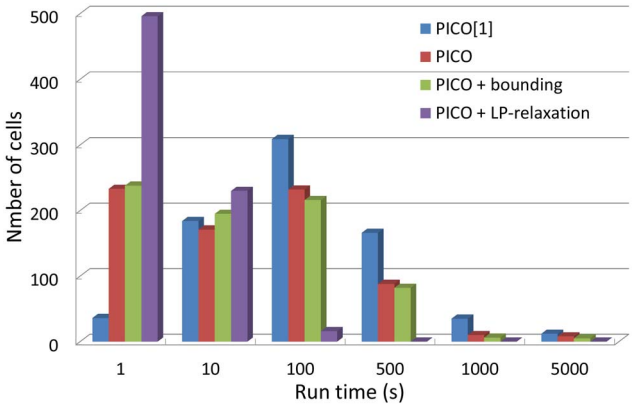

Fig. 15. Run time of PICO algorithm for all cells across the entire cell library.

This further demonstrates the necessity of MILP formulation to maximize the pin access flexibility for the entire SC library.

Since the PAO, including double patterning aware Via-1 assignment and SADP aware pin access design, is implemented at the SC level, the optimization run time for a specific hit point combination is < 0.1 s. However, the majority of the total run time is due to the backtracking method used to enumerate all hit point combinations and the run time of PICO algorithm for all cells across the library is given as a histogram in Fig. 15. We compare the run time from [1] ("PICO [1]"), PICO without check heuristics ("PICO"), PICO with check heuristics (PICO + bounding) and PICO with LP relaxation (PICO + LP-relaxation). The optimization for over 95% of the cells can be finished within 500 s for PICO + bounding. We see significant run time reduction from [1], even after incorporating LELE aware Via-1 assignment, due to the improved memory usage in our coding framework as well as the addition of the MILP solver application programming interface. For example, the optimization of over 200 cells can be finished within 1 s, which is a huge improvement from [1]. In addition, more cells introduce less run time with check heuristics applied. The PICO + LP-relaxation guarantees that the optimization for each cell can be finished within 100 s, although some valid hit point combinations and valid hit points are missed. Since PICO is a one-time computation per cell, it is worthwhile to extend the framework to the entire library to avoid potential engineering efforts related to pin access design without SADP-aware optimization and apply the MILP formulation to maximize the pin access flexibility for the SC library.

D. Guidelines for Routing Stage

The ultimate goal of the PICO problem is to ensure SADP-legal routing by presenting SADP-friendly pin access to the physical design tools, *a priori*. In this paper, we maximize the pin access flexibility at the SC level and then provide the guidance extracted from PICO to the routing stage. Here, we present two possible techniques, including scoring hit points and pin access selection, to transfer the PAO results for the detailed routing tool.

1) Score Hit Points: As discussed previously, there may be multiple hit points for each I/O pin in a SC. Each hit point

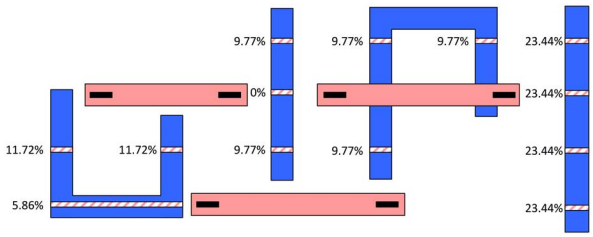

Fig. 16. Hit points scores for a SC.

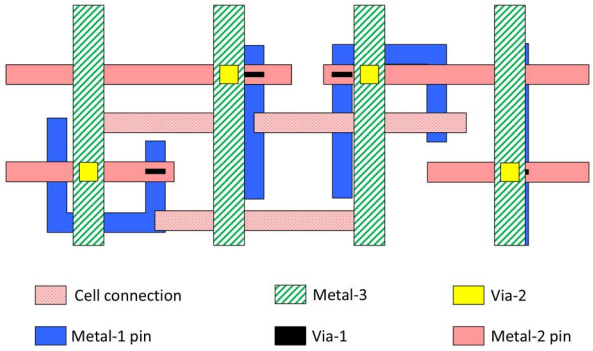

Fig. 17. Pin access selection on the Metal-2 layer for detailed routing.

for an I/O pin influences the other I/O pin hit points in a combination, and can therefore influence the routability of a hit point combination. In order to quantify this effect, we devise a basic scoring strategy for hit points. The elementary scoring strategy for some specific hit point (hp) is given as: score of hp = number of valid hit point combinations associated with hp/total number of hit point combinations enumerated for hp. An example of scoring is shown in Fig. 16. It can be observed that, for the I/O pin on the left, the router prefers two short hit points to the long hit point, due to higher score. The hit point with zero score in the middle reveals that hit point should be blocked for the router.

2) Pin Access Selection: For SADP-aware design, the pin access is the interface between the cell layout design and the place and route tool. Using the proposed methodology, we see significant improvement in the number of valid hit point combinations. If we take the cell robustness, like pin density, and cell placement into consideration, we can identify several robust hit point combinations and report those to physical design tools. As shown in Fig. 17, SADP-aware Metal-2 wires for pin access and cell connection can be optimized and incorporated in the cell layout design. Then, we can promote the Metal-1 pins up to Metal-2 to relieve some of the burden on the router [16].

VII. Conclusion

In this paper, we propose a systematic methodology and introduce two algorithms, PAO for a specific I/O hit point combination and PICO for a SC, which maximize the pin access flexibility for a 10-nm SC library. To the best of our knowledge, this is the first work that addresses LELE-aware Via-1 assignment and SADP-aware I/O pin access design simultaneously. Compared to the conventional approach, we achieve

significant improvement in pin accessibility of our scaled, 10-nm-representative SCs. Our pin access design results also provide maximized flexibility for the routing stage. In the future, we plan to extend the PAO technique to the intercell pin access study considering placement level information. To make use of the pin access flexibility, we also plan to develop novel routing strategies to enable the handshaking between SC level pin access and detailed routing stage.

REFERENCES

[1] X. Xu, B. Cline, G. Yeric, B. Yu, and D. Z. Pan, "Self-aligned double patterning aware pin access and standard cell layout co-optimization," in *Proc. Int. Symp. Phys. Design*, Petaluma, CA, USA, Apr. 2014, pp. 101–108.

[2] B. Yu *et al*., "Dealing with IC manufacturability in extreme scaling," in *Proc. Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, Nov. 2012, pp. 240–242.

[3] M. C. Smayling *et al*., "Sub-12nm optical lithography with 4x pitch division and SMO-lite," *Proc. SPIE*, vol. 8683, Mar. 2013, Art. ID 868305.

[4] D. Z. Pan, B. Yu, and J.-R. Gao, "Design for manufacturing with emerging nanolithography," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 10, pp. 1453–1472, Oct. 2013.

[5] C. A. Mack, *Field Guide to Optical Lithography*, vol. 6. Washington, DC, USA: SPIE, 2006.

[6] H. Zhang, Y. Du, M. D. Wong, and R. Topaloglu, "Self-aligned double patterning decomposition for overlay minimization and hot spot detection," in *Proc. 48th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, New York, NY, USA, Jun. 2011, pp. 71–76.

[7] Y. Ban, K. Lucas, and D. Z. Pan, "Flexible 2-D layout decomposition framework for spacer-type double pattering lithography," in *Proc. 48th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, New York, NY, USA, Jun. 2011, pp. 789–794.

[8] Z. Xiao, H. Zhang, Y. Du, and M. D. Wong, "A polynomial time exact algorithm for self-aligned double patterning layout decomposition," in *Proc. ACM Int. Symp. Int. Symp. Phys. Design (ISPD)*, Napa, CA, USA, Mar. 2012, pp. 17–24.

[9] J.-R. Gao, B. Yu, and D. Z. Pan, "Self-aligned double patterning layout decomposition with complementary e-beam lithography," in *Proc. Asia South Pac. Design Autom. Conf. (ASP-DAC)*, Singapore, Jan. 2014, pp. 143–148.

[10] G. Luk-Pat *et al*., "Avoiding wafer-print artifacts in spacer is dielectric (SID) patterning," *Proc. SPIE*, vol. 8683, Feb. 2013, Art. ID 868312.

[11] M. Mirsaeedi, J. A. Torres, and M. Anis, "Self-aligned double-patterning (SADP) friendly detailed routing," *Proc. SPIE*, vol. 7974, Feb. 2011, Art. ID 79740O.

[12] J.-R. Gao and D. Z. Pan, "Flexible self-aligned double patterning aware detailed routing with prescribed layout planning," in *Proc. ACM Int. Symp. Phys. Design (ISPD)*, Napa, CA, USA, Mar. 2012, pp. 25–32.

[13] C. Kodama *et al*., "Self-aligned double and quadruple patterning-aware grid routing with hotspots control," in *Proc. IEEE/ACM Asia South Pac. Design Autom. Conf. (ASPDAC)*, Yokohama, Japan, Jan. 2013, pp. 267–272.

[14] Y. Du *et al*., "Spacer-is-dielectric-compliant detailed routing for self-aligned double patterning lithography," in *Proc. 50th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Austin, TX, USA, Jun. 2013, pp. 1–6.

[15] T. Nieberg, "Gridless pin access in detailed routing," in *Proc. 48th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, New York, NY, USA, Jun. 2011, pp. 170–175.

[16] M. M. Ozdal, "Detailed-routing algorithms for dense pin clusters in integrated circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 3, pp. 340–349, Mar. 2009.

[17] Y. Ma *et al*., "Self-aligned double patterning (SADP) compliant design flow," *Proc. SPIE*, vol. 8327, Feb. 2012, Art. ID 832706.

[18] H. Zhang, M. D. Wong, and K.-Y. Chao, "On process-aware 1-D standard cell design," in *Proc. 15th Asia South Pac. Design Autom. Conf. (ASP-DAC)*, Taipei, Taiwan, Jan. 2010, pp. 838–842.

[19] H. Zhang, Y. Du, M. D. Wong, and K.-Y. Chao, "Mask cost reduction with circuit performance consideration for self-aligned double patterning," in *Proc. 16th Asia South Pac. Design Autom. Conf. (ASP-DAC)*, Yokohama, Japan, Jan. 2011, pp. 787–792.

[20] A. B. Kahng, C.-H. Park, X. Xu, and H. Yao, "Layout decomposition for double patterning lithography," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, Nov. 2008, pp. 465–472.

[21] K. Yuan, J.-S. Yang, and D. Z. Pan, "Double patterning layout decomposition for simultaneous conflict and stitch minimization," in *Proc. ACM Int. Symp. Phys. Design (ISPD)*, San Diego, CA, USA, Mar./Apr. 2009, pp. 107–114.

[22] Y. Xu and C. Chu, "GREMA: Graph reduction based efficient mask assignment for double patterning technology," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, Nov. 2009, pp. 601–606.

[23] J.-S. Yang, K. Lu, M. Cho, K. Yuan, and D. Z. Pan, "A new graph-theoretic, multi-objective layout decomposition framework for double patterning lithography," in *Proc. 15th Asia South Pac. Design Autom. Conf. (ASP-DAC)*, Taipei, Taiwan, Jan. 2010, pp. 637–644.

[24] X. Tang and M. Cho, "Optimal layout decomposition for double patterning technology," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, Nov. 2011, pp. 9–13.

[25] D. Nenni. (Nov. 4, 2012). *16nm FinFET Versus 20nm Planar!* [Online]. Available: http://www.semiwiki.com/forum/content/1789-16nm-finfet-versus-20nm-planar.html

[26] G. Luk-Pat *et al*., "Design compliance for spacer is dielectric (SID) patterning," *Proc. SPIE*, vol. 8326, Feb. 2012, Art. ID 83260D.

[27] F.-L. Heng, Z. Chen, and G. E. Tellez, "A VLSI artwork legalization technique based on a new criterion of minimum layout perturbation," in *Proc. Int. Symp. Phys. Design*, Napa Valley, CA, USA, Apr. 1997, pp. 116–121.

[28] R. S. Ghaida *et al*., "Layout decomposition and legalization for double-patterning technology," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 2, pp. 202–215, Feb. 2013.

[29] A. Agarwal, S. Bhat, A. Gray, and I. E. Grossmann, "Automating mathematical program transformations," in *Proc. 12th Int. Conf. Pract. Aspects Declarative Lang.*, 2010, pp. 134–148.

[30] (Jan. 11, 2007). *CBC*. [Online]. Available: http://www.coin-or.org/projects/Cbc.xml

Xiaoqing Xu (S'15) received the B.S. degree in microelectronics from Peking University, Beijing, China, in 2012. He is currently pursuing the Ph.D. degree in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, under the supervision of Prof. D. Z. Pan.

His current research interests include robust standard cell design, design for manufacturability, and physical design.

Mr. Xu was the recipient of the Microelectronics and Computer Development Fellowship from the University of Texas at Austin in 2012 and the Excellent Graduate from Peking University in 2012.

Brian Cline (M'10) received the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA.

He joined ARM Holdings, Cambridge, U.K., where he is currently a Principle Design Engineer focusing on logic, memory, and processor design and performance at advanced process nodes.

Greg Yeric (SM'00) received the Ph.D. degree in microelectronics from the University of Texas at Austin, Austin, TX, USA, in 1993.

He joined Motorolas Advanced Products Research and Development Laboratories in embedded non-volatile memory process integration, subsequently researching on technology development roles at TestChip Technologies, Plano, TX, USA, HPL Technologies, San Jose, CA, USA and Synopsys, Mountain View, CA, USA. He is currently a Senior Principal Research and Development Engineer with ARM Holdings, Cambridge, U.K., focusing on design-technology co-optimization and predictive technology.

Bei Yu (S'11–M'14) received the Ph.D. degree from the University of Texas at Austin, Austin, TX, USA, in 2014.

He is currently a Post-Doctoral Scholar with the Department of Electrical and Computer Engineering, University of Texas at Austin. His current research interests include design for manufacturability and optimization algorithms.

Dr. Yu was the recipient of the two Best Paper Awards at the International Conference on Computer Aided Design (ICCAD) in 2013 and the Asia and South Pacific Design Automation Conference (ASPDAC) in 2012, and three other Best Paper Award Nominations at the Design Automation Conference (DAC) in 2014, ASPDAC'13, and ICCAD'11, the Chinese Government Award for Outstanding Students Abroad in 2013, the Society of Photo-Optical Instrumentation Engineers Education Scholarship in 2013, the Silver Medal in Association for Computing Machinery (ACM) Student Research Contest at ICCAD'13, and the IBM Ph.D. Scholarship in 2012.

David Z. Pan (S'97–M'00–SM'06–F'14) received the B.S. degree from Peking University, Beijing, China, and the M.S. and Ph.D. degrees from the University of California, Los Angeles (UCLA), Los Angeles, CA, USA.

From 2000 to 2003, he was a Research Staff Member at IBM T. J. Watson Research Center, Yorktown Heights, NY, USA. He is currently the Engineering Foundation Professor with the Department of Electrical and Computer Engineering, The University of Texas (UT) at Austin, Austin, TX, USA. His current research interests include nanoscale design for manufacturability and reliability, physical design, vertical integration design and technology, and design/CAD for emerging technologies. He has published over 200 papers in refereed journals and conferences, and holds eight U.S. patents.

Mr. Pan was the recipient of several awards for his research contributions and professional services, including the SRC 2013 Technical Excellence Award, the DAC Top Ten Author in Fifth Decade, the DAC Prolific Author Award, the Asia and South Pacific Design Automation Conference Frequently Cited Author Award in 2015, the 11 Best Paper Awards at premier venues, such as International Symposium on Physical Design (ISPD)'14, the International Conference on Computer Aided Design (ICCAD) 2013, the Asia and South Pacific Design Automation Conference (ASPDAC) 2012, ISPD'11, the IBM Research 2010 Pat Goldberg Memorial Best Paper Award in Computer Science/Electrical Engineering/Math, ASPDAC'10, Design, Automation & Test in Europe'09, International Conference on IC Design and Technology'09, and the SRC Techcon in 1998, 2007, and 2012, the Communications of the ACM Research Highlights in 2014, the ACM/SIGDA Outstanding New Faculty Award in 2005, the NSF CAREER Award in 2007, the SRC Inventor Recognition Award thrice, the IBM Faculty Award four times, the UCLA Engineering Distinguished Young Alumnus Award in 2009, the UT Austin RAISE Faculty Excellence Award in 2014, the ISPD Routing Contest Awards in 2007, the eASIC Placement Contest Grand Prize in 2009, ICCAD'12 and ICCAD'13, the CAD Contest Awards, the IBM Research Bravo Award in 2003, the Dimitris Chorafas Foundation Research Award in 2000, and the ACM Recognition of Service Award in 2007 and 2008. From 2008 to 2009, he was an IEEE CAS Society Distinguished Lecturer. He was a Senior Associate Editor of *ACM Transactions on Design Automation of Electronic Systems*, an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II, *Science China Information Sciences*, the *Journal of Computer Science and Technology*, and the IEEE CAS Society Newsletter. He has served as the Chair for the IEEE Computer Aided Network DEsign Committee and the ACM/SIGDA Physical Design Technical Committee, a Program/General Chair for ISPD, Technical Program Committee Subcommittee Chair for DAC, ICCAD, ASPDAC, International Symposium on Low Power Electronics and Design, International Conference on Computer Design, International Symposium on Circuits and Systems, International Symposium on VLSI Design, Automation & Test, International Symposium on Quality Electronic Design, a Tutorial Chair for DAC 2014, and the Workshop Chair for ICCAD 2015.