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

Xiaoqing Xu, Brian Cline†, Greg Yeric†, Bei Yu, David Z. Pan
ECE Department, Univ. of Texas at Austin, Austin, TX, USA
†ARM Inc, Austin, TX, USA
{xiaoqingxu, bei, dpan}@cerc.utexas.edu, {Brian.Cline, Greg.Yeric}@arm.com

ABSTRACT

Self-Aligned Double Patterning (SADP) is being considered for use at the 10nm technology node and below for routing layers with pitches down to ∼50nm because it has better LER 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. In this paper, we present the first study on SADP-aware pin access and layout optimization at the SC level. Accounting for SADP-specific design rules, we propose a coherent framework that uses Mixed Integer Linear Programming (MILP) and branch and bound method to 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 standard cells and maximizes the pin access flexibility for routing.

Categories and Subject Descriptors

B.7.2 [**Hardware, Integrated Circuit**]: Design Aids

General Terms

Algorithms, Design, Performance

Keywords

Self-Aligned Double Patterning (SADP), Pin Access, Standard Cell Layout

1. INTRODUCTION

Due to the resolution limits of 193nm photolithography, double patterning techniques and regular layout have been widely used to extend semiconductor process technology scaling [1–3]. 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 >20nm. 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 standard cell (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 14nm technologies and below. That means automated standard cell layout design and optimization are needed to provide flexible I/O pin access.

SADP, in particular, is a viable candidate for lower layer metallization with regular patterns at the 10nm technology node, due to better overlay and Line Edge Roughness (LER) 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 [4–7]. For regular layout, the line-space array decomposition method can efficiently decompose SADP-based geometries and achieve good pattern fidelity and process margin [2], [8]. 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

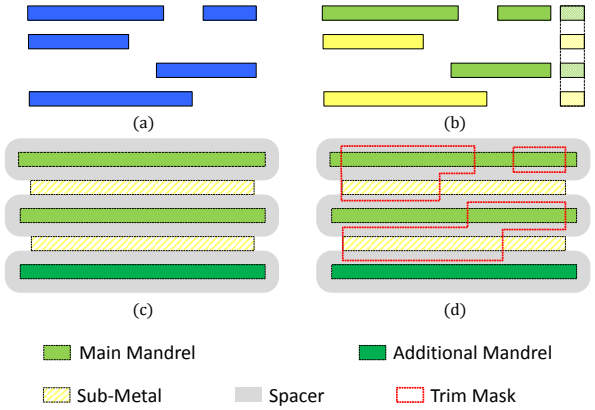

Figure 1: The line-space array decomposition, (a) target layout, (b) layout coloring, (c) mandrel mask design, (d) trim mask design.

ISPD'14, March 30–April 2, 2014, Petaluma, CA, USA.

http://dx.doi.org/10.1145/2560519.2560530.

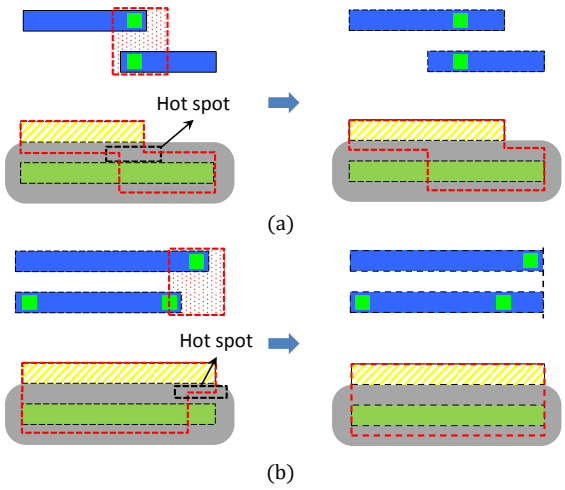

Figure 2: Line-end extension techniques, (a) anti-parallel line ends, (b) parallel line ends.

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 [9–12], 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. 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 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 [13].

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.

In this work we formulate this issue as an SC I/O pin access problem and illustrate the usefulness of our methodology at the 10nm technology node. To solve this problem efficiently, this paper proposes a Mixed Integer Linear Programming (MILP) based technique that simultaneously optimizes the Metal-2 wires used for pin access and within-cell connections of standard cells. In addition, using the branch and bound method, we extend this technique to each pin access strategy and maximize pin accessibility for each cell in the $10nm$ library. Our main contributions are summarized as follows:

- To the best of our knowledge, this is the first work that addresses SC I/O pin access design at the SC level.
- We propose a MILP-based optimization methodology to enable SADP-aware Metal-2 layout design for pin access and within-cell connections.
- The pin access and cell layout co-optimization is proposed to systematically maximize the pin access flexibility for the entire standard cell library.

The rest of this paper is organized as follows: Section 2 introduces background material relevant to the pin access design issue. Section 3 shows the formulation of the SADP-aware pin access design problem, including several definitions and our design target. Section 4 presents our MILP-based methodology for Metal-2 layout optimization. Section 5 extends our optimization technique to the entire standard cell library based on the branch and bound method. Section 6 demonstrates the effectiveness of our optimization framework, and compares the SADP-aware pin access optimization to the conventional approach with the design rule checks. Section 7 concludes this paper.

2. PRELIMINARIES

2.1 Line-Space Array Decomposition

The continued geometric scaling of process technology depends on multiple patterning and increased layout regularity [2]. Thus, at the 10nm node and beyond, we assume that the Metal-2 layout will be extremely regular. Furthermore, in the 10nm 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 $10nm$ technology node used, the Metal-2 pitch was assumed to be $M2_{pitch} = 48nm$, which corresponds to a 75% scaling from the $16nm$ node [14]. Thus, the pitch between even/odd routing tracks was $96nm$ ($2 \times M2_{pitch}$), which was larger than the resolution limit of $193nm$ 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 [8] from Fig. 1 can be deployed. Basically, each wire on the same routing track is extended and merged to achieve the line-space array. As illustrated in Fig. 1(c), we place mandrel features on non-adjacent lines and add additional mandrels when necessary. In the *Spacer Is Dielectric* (SID) 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 towards the 10nm 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

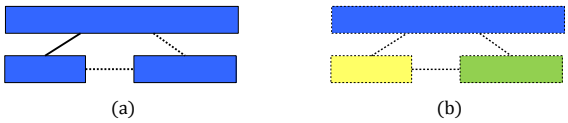

Figure 3: The potential odd-cycle conflicts of the coloring graph, (a) potential Metal-2 layout, (b) potential odd-cycle conflicts.

Table 1: 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

time, which leads to the layout patterns illustrated in Fig. 3. In Fig. 3(a), a solid edge denotes coloring conflict and a dashed edge denotes a potential coloring conflict. Fig. 3(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 [5]. The color assignment strategy from observation 2 helps to avoid potential odd cycles in the coloring stage for SADP-friendly layout.

2.2 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 4 design rules shown in Fig. 4 that enforce SADP-friendly layout using 1-D relationships.

Table 1 defines SADP-specific notations for the 10nm technology node [15]. 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. 4(a). We use **OffTrackOverlap(l_2)** and **OffTrackSpace(l_3)** to define the prohibited region for the anti-parallel Metal-2 wires [15], as shown in Fig. 4(b) and Fig. 4(c), respectively. Finally, for parallel Metal-2 wires illustrated in Fig. 4(d), the line-end design constraint is defined as **OffTrackOffset(l_4)**. Table 2 summarizes the design rules for Metal-2 layout patterns [13, 15].

3. PROBLEM FORMULATION

Our I/O pin access on the Metal-2 layer is based on practical layout of standard cells. 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 SADP-aware pin access design problem, we have the following definitions.

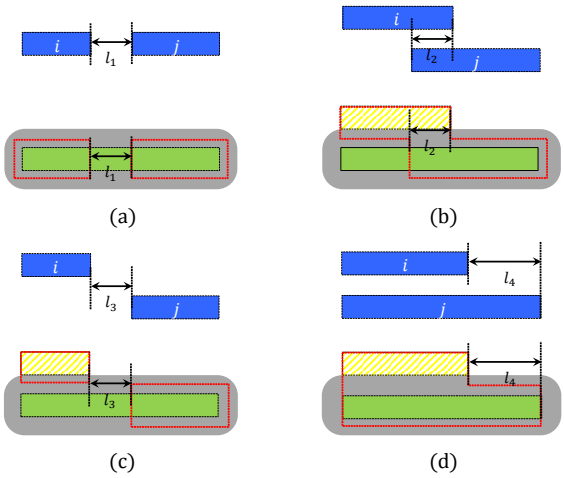

Figure 4: SADP-specific design rule formulation, (a) $OnTrackSpace \geq l_1$, (b) $OffTrackOverlap \geq l_2$, (c) $OffTrackSpace \geq l_3$, (d) $OffTrackOffset \geq l_4$ or $OffTrackOffset = l_4 = 0$.

Table 2: 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	$l_4 \geq w_r$ or $l_4 = 0$

Definition 1 (Hit Point) *The intersection of a Metal-2 routing track (which is pre-determined 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 $10nm$ technology used, Metal-2 is uni-directional 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 need to design the Metal-2 wire assuming one accessing direction for the hit point. Then, 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.*

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.*

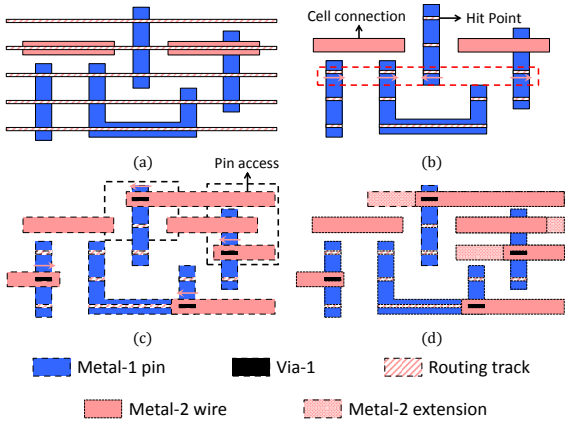

Figure 5: A simple example for pin access design, (a) Metal-2 tracks and cell layout, (b) hit points for each I/O pin, (c) Metal-2 wires for pin access, (d) line end extension.

A simple example of the I/O pin access design for one hit point combination is shown in Fig. 5. Fig. 5(a) demonstrates the Metal-1 I/O pin layout and Metal-2 routing tracks running horizontally above Metal-1. Fig. 5(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 decided by the minimum width of 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. 5(b) shows one hit point combination and the pink arrows denote the arbitrary accessing directions chosen for the hit points. Fig. 5(c) illustrates another hit point combination and 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, 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. 5(c) denote all pairs of line ends that cause hot spots in trim mask designs. Fig. 5(d) demonstrates that we can make use of line-end extension techniques to fix those hot spots in the trim mask.

However, in SC design, it is non-trivial 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 design the Metal-2 wires for pin access and within-cell connections simultaneously. We can now define the SADP-aware pin access optimization problem as follows.

Problem 1 (Pin Access Optimization (PAO)) *Given the standard cell layout and a specific hit point combination, determine whether or not it is possible to optimize the Metal-2 wires for pin access and within-cell connections under SADP constraints. If possible, show all SADP-friendly Metal-2 wires.*

Moreover, as shown in Fig. 5(b), we may have multiple hit points for one I/O pin, which leads to numerous hit point combinations for one cell. For one standard cell, we define the pin access and cell layout co-optimization problem as follows.

Table 3: 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

Problem 2 (PICO) *Given the standard cell layout, the Pin Access and Cell Layout Co-Optimization (PICO) problem is to show all Metal-2 wiring cases with successful PAO's and maximize the pin access flexibility under SADP constraints.*

4. SADP-AWARE PIN ACCESS

Given a specific hit point combination, we pre-design the Metal-2 wires for pin access. Then, we propose an MILP-based method to solve PAO problem efficiently.

4.1 Pin Access Pre-design

Given a hit point combination and the accessing direction for each hit point, we can determine the position of Via-1 and extend the line ends of the Metal-2 wires for pin access accordingly. Specifically, if we need to access the hit point from the right of cell, we will put the Via-1 as close to the right of the hit point as possible. Then, we can determine the line end position of the corresponding 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 standard cell boundary. Hence, if one hit point is next to the right boundary of the 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 pre-design if the hit point is accessed from the left boundary of cell. Fig. 5(c) is an example of Metal-2 wires for pin access after the pre-design stage. The pre-design method induces the following observation.

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

4.2 Pin Access Optimization

As illustrated in Fig. 5(c), if we simply use the hit point to determine the line end of Metal-2 wires, the SADP constraints may invalidate some hit point combinations. The line end extension techniques motivate us to legalize the Metal-2 layout to enable SADP-friendly design. The conventional layout migration issue has been formulated as a linear programming problem in [16]. A similar approach has also been deployed to deal with LELE double patterning layout decomposition in [17]. However, the linear programming technique used in [16] and [17] 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 3 shows the notations for the variables used in our formulation. We will first give the mathematical formulation for our Pin Access Optimization (PAO) 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.

4.2.1 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 [13]. 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 pin access optimization. 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 2) 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 j_{th} wire, as demonstrated in Fig. 4(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 *Rule* 2 or *Rule* 3. Then, constraints ($1h$) and ($1i$) are formulated to specify *Rule* 4 for each pair of Metal-2 wires in set S_4.

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

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

4.2.2 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 [18].

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.

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

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

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

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 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.

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

5. PIN ACCESS AND CELL LAYOUT CO-OPTIMIZATION

Previously, we have shown that MILP-based optimization determines whether a single hit point combination is valid or not. If it is valid, we can achieve successful optimization of Metal-2 wires for pin access and cell connection simultaneously. However, as shown in Fig. 5, multiple hit points for one I/O pin lead to numerous hit point combinations for one standard cell. In general, the more valid hit point combinations we have for one cell, the more flexibility we can provide to the routing stage. Thus, we extend the MILP-based optimization to validate all hit point combinations of a standard cell.

The overall algorithm for the pin access and cell layout co-optimization (PICO) is given in Algorithm 1. First, as shown in lines 1-7, the preprocessing steps determine the set of hit points for each I/O pin. Then, in line 9, the branch and bound method is proposed to obtain a table of potential valid hit point combinations for the standard cell. From line 10 to line 15, we call the MILP-based optimization (2) for each entry in the table of potential hit point combinations. For each cell, all valid pin access designs are stored in a table, which can be incorporated into the standard cell library

design. Hence, we have maximized the pin access flexibility of one cell for the routing stage.

Algorithm 1 PICO Algorithm

Require: Cell layout and Metal-2 routing tracks;
1: define C as set of Metal-2 wires for cell connection;
2: define IO as set of I/O pins;
3: define S_{IO} as the 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_{IO};
7: **end for**
8: define $MTable$ as table of Metal-2 layout design;
9: $MTable$ = Branch-and-Bound(S_{IO}, C);
10: **for** each entry H_k in $MTable$ **do**;
11: **if** the pin access optimization for $H_k \cup C$ is feasible **then**
12: replace H_k with the feasible solution;
13: **else**
14: delete H_k;
15: **end if**
16: **end for**

Algorithm 2 illustrates the branch and bound method that yields a table of potential valid hit point combinations. In line 2, we construct a search tree out of all hit points for each I/O pin. Fig. 6 demonstrates the construction of the search tree from all hit points (p_i^k) of any standard cell. The root will be a virtual node for the tree. A path from root to leaf gives a hit point combination for the cell. The i_{th} level of the search tree contains the hit points for the i_{th} I/O pin of the cell. Lines 4-14 demonstrate the process of table construction. While traversing from root to node, before adding a new node to path, we should check the compatibility of this node with its ancestor nodes in path and Metal-2 wires for within-cell connections. We continue the traversal if the node is legal on path. Otherwise, we go to next node on the same level, which prunes the redundant hit point combinations relevant to that node.

Algorithm 2 Branch-and-Bound Algorithm

Require: a set of hit points (S_{IO}) and a set of Metal-2 wires (C);
1: **function** BRANCH-AND-BOUND(S_{IO}, C)
2: construct search tree based on S_{IO};
3: define list $Path$ for traversed nodes;
4: **repeat**
5: $Path = \emptyset$;
6: **while** traverse from root to leaf **do**
7: **if** Check(*currentnode*, $Path$, C)=True **then**
▷ The check heuristics for branch and bound
8: push(*currentnode*, $Path$);
9: **else**
10: go to next node on the same level;
11: **end if**
12: **end while**
13: add $Path$ to $Table$
14: **until** all paths exhausted
15: return $Table$;
16: **end function**

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

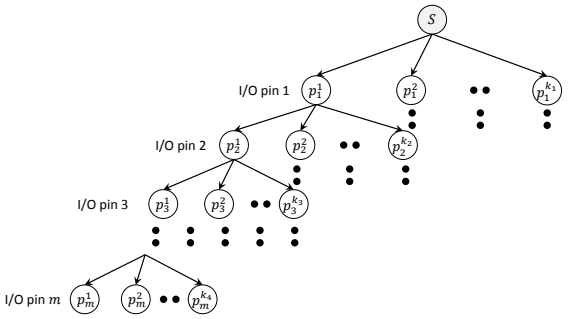

Figure 6: Search tree for the branch and bound method

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.

It shall be noted that we can further consider other pruning metrics during the branch and bound stage. For example, cell robustness metrics, such as pin density, are closely related to the pin access design at the standard cell level and it could be another metric used to prune out invalid hit point combinations.

6. EXPERIMENTAL RESULTS

We have implemented our algorithm in C++ and tested it using an industrial $14nm$ standard cell library that has been scaled and compacted to $10nm$-representative dimensions. We use CBC [19] as our MILP solver and all experiments are performed on a Linux machine with 3.33GHz Intel(R) Xeon(R) CPU X5680. The width and space of Metal-2 wires are assumed to be $24nm$. The spacer deposit width is set as $24nm$. For trim mask design, the minimum resist width and space are set as $44nm$ and $46nm$, respectively. The etch bias is set as $6nm$ [15].

Next, we demonstrate the strength of our optimization methodology by showing the results from pin access design for specific hit point combinations, standard cells and the entire standard cell library consisting of around 700 cells. Fig. 7 demonstrates a typical cell layout design in the 10nm technology node. 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. 7(a) shows the Metal-2 layout design if we simply use hit point (Via-1) locations to determine the line end positions. A design rule check will reveal multiple violations in the dashed boxes. However, as illustrated in Fig. 7(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.

The PICO problem motivates us to extend the optimization technique to each hit point combination of the standard cell based on the branch and bound method. The pin access flexibility of each cell can be evaluated in terms of the number of hit point combinations or valid hit points for each I/O pin. As shown in Fig. 7, we may have various hit points for each I/O pin of the cell. We assume that

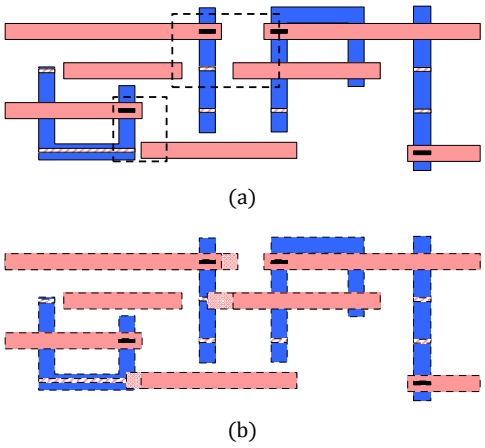

Figure 7: Pin access and cell connection co-optimization for one hit point combination. (a) design rule violations in layout, (b) MILP-based optimization result.

the hit point 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. Fig. 8 shows the effectiveness of the PICO for 5 typical cells in the 10nm standard cell 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 SADP constraints but there is no significant improvement in terms of hit point combinations. However, we achieve a significant increase in the number of valid hit point combinations for other cells. Particularly, conventional design rule checking invalidates all of the hit point combinations for Cell 5, the layout of which is shown in Fig. 7. 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.

For both conventional and SADP-aware pin access design, we further evaluate the number of valid hit points based on the definition of the valid hit point shown in Section 3. Compared to the conventional approach, the SADP-aware pin access design achieves the increase of valid hit points for each I/O pin of the cells, as demonstrated in Table 4. 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 3 I/O pins, so pins #4 and #5 have a "-" in their entry 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 4, but show significant increases in the number of valid hit point combinations in Fig. 8. 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.

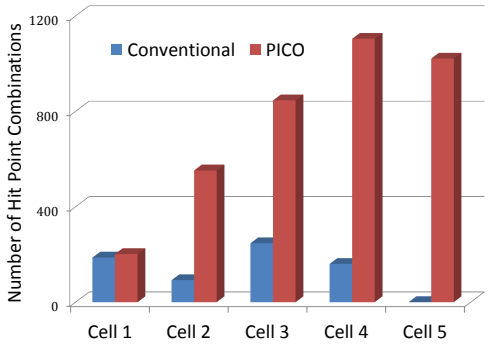

Figure 8: The increase in the number of valid hit point combinations

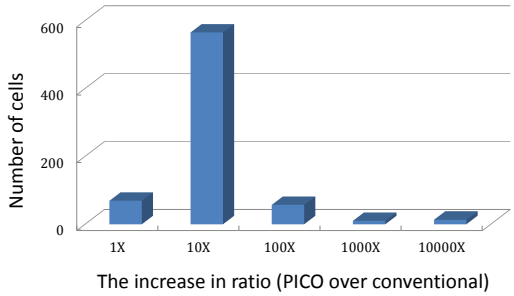

Figure 9: The increase in ratio on the number of valid hit point combinations across the entire cell library

To gauge the library-wide effectiveness of our optimization framework, we also applied the proposed technique to each cell in our 10nm library. We calculated the ratio of valid hit point combinations of PICO algorithm over the conventional approach for each cell. The histogram in Fig. 9 demonstrates the valid hit point combination ratio and the effectiveness of PICO technique. We obtain 10X or more improvement for most cells and some cells achieve up to a 10000X increase in the number of valid hit point combinations. This means that the PICO has significantly improved the pin access flexibility for the routing stage. We also evaluate the increase in the number of valid hit points for each I/O pin. The increase in percentage over total number of hit points is calculated for each cell in the library and shown in Fig. 10. In our 10nm experiments, we find that over 25 % of cells have 20 % improvement in the number of valid hit points.

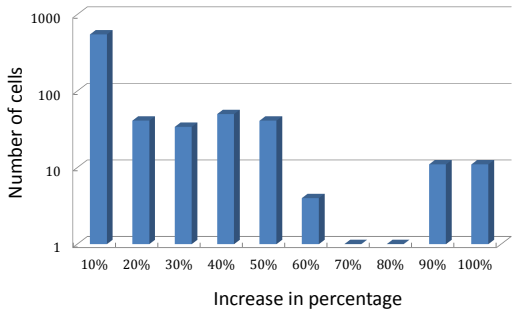

Figure 10: The increase in percentage on the number of valid hit points across the entire cell library

Table 4: The increase in the number of valid hit points for each I/O pin

	Conventional					PICO				
pin index	#1	#2	#3	#4	#5	#1	#2	#3	#4	#5
Cell 1	2	5	3	-	-	2	5	3	-	-
Cell 2	1	0	0	4	-	1	3	3	4	-
Cell 3	2	3	2	3	3	2	3	2	3	3
Cell 4	1	0	0	1	3	1	3	3	1	4
Cell 5	0	0	0	0	-	3	2	2	5	-

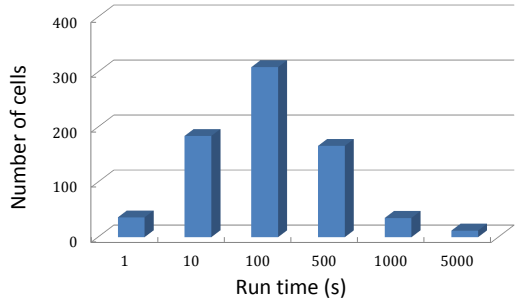

Figure 11: The run time of PICO algorithm for all cells across the entire cell library

Since the MILP-based optimization is implemented at the standard cell level, the optimization runtime for a specific hit point combination is < 0.1 sec. However, the majority of the total runtime is due to the branch and bound method used to enumerate all hit point combinations and the runtime of PICO algorithm for all cells across the library is given as a histogram in Fig. 11. For most cells in the library, the optimization can be finished within 500 sec. 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.

7. 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 standard cell, which maximize the pin access flexibility for a $10nm$ standard cell library. To the best of our knowledge, this is the first work that addresses SADP-aware I/O pin access design. Compared to the conventional approach, we achieve significant improvement in pin accessibility of standard cells for the 10nm technology node, which is likely to use SADP for Metal-2 routing. Our pin access design results also provide maximized flexibility for the routing stage.

8. ACKNOWLEDGEMENTS

This work is supported in part by NSF and SRC.

9. REFERENCES

[1] B. Yu, J.-R. Gao, D. Ding, Y. Ban, J.-s. Yang, K. Yuan, M. Cho, and D. Z. Pan, "Dealing with IC Manufacturability in Extreme Scaling," in *Proceedings of the International Conference on Computer-Aided Design*. ACM, 2012, pp. 240–242.

[2] M. C. Smayling, K. Tsujita, H. Yaegashi, V. Axelrad, T. Arai, K. Oyama, and A. Hara, "Sub-12nm optical lithography with 4x pitch division and SMO-lite," in *SPIE Advanced Lithography*. International Society for Optics and Photonics, 2013, pp. 868 305–868 305.

[3] D. Z. Pan, B. Yu, and J.-R. Gao, "Design for Manufacturing With Emerging Nanolithography," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 32, no. 10, pp. 1453–1472, 2013.

[4] H. Zhang, Y. Du, M. D. Wong, and R. Topaloglu, "Self-Aligned Double Patterning Decomposition for Overlay Minimization and Hot Spot Detection," in *IEEE/ACM Design Automation Conference (DAC)*, 2011.

[5] Y. Ban, K. Lucas, and D. Z. Pan, "Flexible 2D Layout Decomposition Framework for Spacer-type Double Pattering Lithography," in *IEEE/ACM Design Automation Conference (DAC)*, 2011, pp. 789–794.

[6] Z. Xiao, H. Zhang, Y. Du, and M. D. Wong, "A Polynomial Time Exact Algorithm for Self-Aligned Double Patterning Layout Decomposition," in *ACM International Symposium on Physical Design (ISPD)*, 2012.

[7] J.-R. Gao, B. Yu, and D. Z. Pan, "Self-aligned Double Patterning Layout Decomposition with Complementary E-Beam Lithography," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2014.

[8] G. Luk-Pat, B. Painter, A. Miloslavsky, P. De Bisschop, A. Beacham, and K. Lucas, "Avoiding wafer-print artifacts in spacer is dielectric (SID) patterning," in *SPIE Advanced Lithography*. International Society for Optics and Photonics, 2013, pp. 868 312–868 312.

[9] M. Mirsaeedi, J. A. Torres, and M. Anis, "Self-aligned double-patterning (SADP) friendly detailed routing," in *Proc. of SPIE*, vol. 7974, 2011.

[10] J.-R. Gao and D. Z. Pan, "Flexible Self-aligned Double Patterning Aware Detailed Routing with Prescribed Layout Planning," in *ACM International Symposium on Physical Design (ISPD)*, 2012.

[11] C. Kodama, H. Ichikawa, K. Nakayama, T. Kotani, S. Nojima, S. Mimotogi, S. Miyamoto, and A. Takahashi, "Self-Aligned Double and Quadruple Patterning-Aware Grid Routing with Hotspots Control," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2013.

[12] Y. Du, Q. Ma, H. Song, J. Shiely, G. Luk-Pat, A. Miloslavsky, and M. D. Wong, "Spacer-is-dielectric-compliant detailed routing for self-aligned double patterning lithography," in *Proceedings of the 50th Annual Design Automation Conference*. ACM, 2013, p. 93.

[13] Y. Ma, J. Sweis, H. Yoshida, Y. Wang, J. Kye, and H. J. Levinson, "Self-aligned double patterning (SADP) compliant design flow," in *SPIE Advanced Lithography*. International Society for Optics and Photonics, 2012, pp. 832 706–832 706.

[14] D. Nenni, "16nm FinFET versus 20nm Planar!" http://www.semiwiki.com/forum/content/1789-16nm-finfet-versus-20nm-planar.html.

[15] G. Luk-Pat, A. Miloslavsky, B. Painter, L. Lin, P. De Bisschop, and K. Lucas, "Design compliance for spacer is dielectric (SID) patterning," in *SPIE Advanced Lithography*. International Society for Optics and Photonics, 2012, pp. 83 260D–83 260D.

[16] F.-L. Heng, Z. Chen, and G. E. Tellez, "A VLSI artwork legalization technique based on a new criterion of minimum layout perturbation," in *Proceedings of the 1997 international symposium on Physical design*. ACM, 1997, pp. 116–121.

[17] R. S. Ghaida, K. B. Agarwal, S. R. Nassif, X. Yuan, L. W. Liebmann, and P. Gupta, "Layout Decomposition and Legalization for Double-Patterning Technology," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 32, no. 2, pp. 202–215, 2013.

[18] A. Agarwal, S. Bhat, A. Gray, and I. E. Grossmann, "Automating mathematical program transformations," in *Practical Aspects of Declarative Languages*. Springer, 2010, pp. 134–148.

[19] "Cbc," http://www.coin-or.org/projects/Cbc.xml.