MILP-Based Optimization of 2-D Block Masks for Timing-Aware Dummy Segment Removal in Self-Aligned Multiple Patterning Layouts

Peter Debacker, *Student Member, IEEE*, Kwangsoo Han, *Student Member, IEEE*, Andrew B. Kahng, *Fellow, IEEE*, Hyein Lee, *Student Member, IEEE*, Praveen Raghavan, *Member, IEEE*, and Lutong Wang, *Student Member, IEEE*

***Abstract*—Self-aligned multiple patterning, due to its low overlay error, has emerged as the leading option for 1-D gridded back-end-of-line (BEOL) in sub-14-nm nodes. To form actual routing patterns from a uniform "sea of wires," *cut masks* are needed for line-end cutting or realization of space between routing segments. The line-end cutting results in nonfunctional (i.e., dummy fill) patterns that change wire capacitance, and hence design timing and power. Therefore, to remove such dummy fill patterns, extra 2-D *block masks* are used. However, 2-D block masks cannot remove arbitrary dummy fill patterns, due to design rule constraints on the block mask shapes. In this paper, we address the *timing-aware* optimization of 2-D block mask layouts under various sets of mask rules that are derived from mask patterning technology options (e.g., 193i and 193d) for foundry 7-/5-nm (N7/N5) BEOL. Our central contribution is a mixed integer linear programming (MILP) optimization that minimizes timing impact due to dummy metal segments while satisfying block mask rules and metal density constraints. We also propose a distributed optimization flow to improve the scalability. With our optimizer, we recover up to 84% of the worst negative slack impact from dummy segments, with up to 64% dummy removal rate. We further extend our MILP to a *co-optimization* of cut and block masks. This paper gives new insights into fundamental limits of benefit from emerging cut and block mask technology options.**

***Index Terms*—Block mask, co-optimization, cut mask, self-aligned multiple patterning (SAMP), timing-aware.**

I. Introduction

SELF-ALIGNED multiple patterning (SAMP), due to its low overlay error, has emerged as the leading option for 1-D gridded back-end-of-line (BEOL) layers in sub-14-nm nodes. To form actual routing patterns from a uniform "sea of wires," *keep*[1] or *block*[2] approaches can be used. The work of [5] demonstrates that mask shapes used to keep signal wire segments (M2 pitch = 32 nm [12], [13]) are not patternable with single-exposure (SE) lithography, even if we assume aggressive optical proximity correction. To address this problem, the block approach is used, wherein both 1-D *cut masks* and 2-D *block masks* are required. 1-D *cut masks* are needed for line-end cutting or realization of space between routing segments, resulting in end-of-line (EOL) extensions and nonfunctional (i.e., dummy fill) patterns.[3]

Despite previous works [1], [3], [7], [15] proposing cut mask optimizations to minimize the EOL extension, such effects as increased capacitance, degraded timing and power are inevitable due to dummy fill patterns. Therefore, extra 2-D *block masks* can be used to remove dummy fill patterns. However, using only 2-D block masks cannot realize line ends due to required complex shapes, particularly with metal pitch $\leq$32 nm in N7/N5 node, which is our focus in this paper. Gillijns *et al.* [5] showed that 2-D block mask shapes fail to realize $\leq$80-nm tip-to-tip spacing between line ends while a 1-D cut mask strategy can realize 56-nm tip-to-tip spacing. Thus, 1-D cut masks are needed to define clean line ends with small tip-to-tip spacing. In this paper, we assume that the cut mask is used to define EOL, and the block mask is used to remove dummy fill patterns or define EOL with a margin.

Fig. 1 illustrates 1-D SAMP patterning with cut and block masks. For a given post-route layout, a "sea of wires" is generated and line ends are defined by a cut mask, as shown in Fig. 1(a) and (b). After the cut process, Fig. 1(c) shows one EOL extension and three nonfunctional dummy segments. 2-D block mask application is shown in Fig. 1(d), and Fig. 1(e) shows the final layout with one EOL extension and one dummy segment. Compared to the layout in Fig. 1(c), Fig. 1(e) is superior with smaller capacitance, lower power, and better timing, due to fewer dummy segments.

Manuscript received July 23, 2016; revised November 16, 2016 and January 17, 2017; accepted February 26, 2017. Date of publication March 21, 2017; date of current version June 16, 2017. This paper was recommended by Associate Editor C. C.-N. Chu.

P. Debacker and P. Raghavan are with IMEC, 3001 Leuven, Belgium (e-mail: peter.debacker@imec.be; praveen.raghavan@imec.be).

K. Han, H. Lee, and L. Wang are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: kwhan@ucsd.edu; hyeinlee@ucsd.edu; luw002@ucsd.edu).

A. B. Kahng is with the Department of Computer Science and Engineering, and Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: abk@ucsd.edu).

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.2017.2685594

[1] *Keep* refers to a mask to keep signal wire segments.

[2] *Block* refers to a mask to erase dummy wire segments.

[3] In terms of layout patterns, cut mask and block mask would act the same since both masks remove unnecessary metal patterns. Indeed, the terms cut and block are used interchangeably in many previous works. In this paper, we use the term cut mask to refer to a 1-D shaped mask, and the term block mask to refer to a 2-D shaped mask.

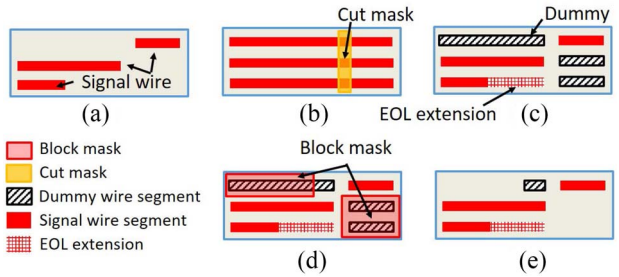


Fig. 1. SAMP process. (a) Post-route layout. (b) Cut mask application. (c) Layout after cut mask application. (d) Block mask application. (e) Final layout after block mask application.

For printability, 2-D block masks must satisfy given design rules from a particular patterning technology. Possible patterning technology options include SE 193i, SE 193d, and extreme ultraviolet lithography (EUV). Except in the case of EUV, the critical dimension for block mask shapes is $\sim 2\times$ larger than the minimum pitch of 1-D SAMP BEOL process. Thus, it is not possible to cover all dummy segments using one block mask. For example, in Fig. 1(e), the two dummy segments in the final layout cannot be removed because of: 1) the minimum spacing constraint between individual block mask shapes and 2) the L-shape constraint. The first main contribution of this paper is that we formulate and optimally solve for 2-D block mask shapes based on realistic design rules of SE 193i and SE 193d patterning technology from industry [21], and with support for a "selective"[4] variant of block-mask patterning technology.

In advanced nodes, minimum metal density is crucial to chemical-mechanical polishing (CMP) [6]. In 1-D SAMP manufacturing process, metal fills are generated intrinsically by the sea-of-wires with cut process, and partially removed by the block mask, as opposed to a dedicated post-routing metal fill process in the traditional physical design flow. Thus, block mask optimization must be metal density-aware.[5] The another contribution of this paper is that we consider the local minimum metal density constraint.

From a performance perspective, maximizing the block mask usage (dummy removal) is not equivalent to minimizing timing impact of dummy fill patterns. In our preliminary study, a timing-oblivious block mask optimization that simply maximizes dummy removal (design: ARM Cortex M0) can only recover 14% of the worst negative slack (WNS) degradation caused by nonfunctional dummy fill patterns. At the same time, timing-aware block mask optimization can run much faster than timing-oblivious optimization since we do not need to optimize nonfunctional dummy fills. Further, given minimum metal density constraints, smart dummy removal method is required to maximize timing recovery. Thus, it is important to capture timing impact of dummy segments in block mask optimization. The second main contribution of this paper is that we incorporate into our optimization a timing model to evaluate dummy fill performance impact. Together with our first main contribution, this enables quantified assessment of performance benefits from selective block mask technology.

Lastly, we extend our mixed integer linear programming (MILP) to a *co-optimization* of cut and block masks, opening up a broader solution space. Compared to a sequential cut [7] and block mask optimization, where line-end realization is performed with cut mask only, a cut and block mask co-optimization seeks to use both cut and block masks for realization of line ends: the block mask can complement the cut mask when a cut-only solution may result in excessive EOL extensions.

To summarize, in this paper we propose an MILP-based optimization for 2-D block mask with timing-aware dummy segment removal, while satisfying a given set of block mask rules (including for selective block mask technology) and metal density constraints. We further provide what we believe to be the first co-optimization of cut and block mask patterns. Our key contributions are as follows.

1) To our knowledge, this paper is the first to optimize 2-D block mask layout considering realistic block mask rules, timing impact of dummy fills and metal density constraints.
2) We develop a timing model to evaluate performance impact on a per-segment basis.
3) We develop a co-optimization of cut and block mask layout.
4) We study the impacts of timing-awareness and patterning technology on optimization outcomes, and we furthermore quantify the power and timing benefits of the "selective" approach.
5) Our MILP formulation gives new insights into fundamental limits of benefit from emerging (cut and) block mask technology options.

In the remainder of this paper, Section II provides background of cut and block mask technology, as well as related work. In Section III, we describe our MILP-based optimization of 2-D block masks and our cut and block mask co-optimization. We also explain our model to capture the timing impact of dummy segments. We describe our conflict list generation techniques, distributed optimization strategy and overall flow in Section IV. Section V provides experimental results and analysis. We give the conclusions and future research directions in Section VI.

II. Background

In this section, we first describe block mask rules and the "selective" block approach. We then review cut mask rules, the selective cut approach, and litho-etch-litho-etch (LELE) cuts. Last, we review relevant related works.

A. Block Mask Rules/Selectivity

Block mask rules constrain each individual shape on the block mask, as well as sets of adjacent block mask shapes. A set of essential rules for block mask shapes is shown in Fig. 2.

[4]That is, a block mask approach that selectively removes metal lines according to the colors of metal (see Section II-A for the detailed description).

[5]Regarding the feasibility of the final pattern after dummy removal in terms of lithography, we note that [5] validates the cut and block mask approach with lithography simulation. We also note that CMP effects from pattern density occur at relatively large length scales compared to feature and pitch dimensions in N7/N5 Mx layer.

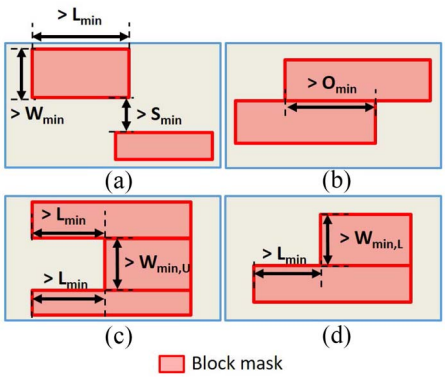


Fig. 2. Block mask rules. Minimum (a) width and length rules, (b) overlap rule, (c) U-shape rule, and (d) L-shape rule.

TABLE I
PRELIMINARY CUT AND BLOCK MASK RULES

Rule	Notation	Meaning	Values (nm)	
			193i	193d
R1	W_{min}	minimum width	60	120
R2	S_{min}	minimum spacing	240	480
R3	L_{min}	minimum length	120	240
R4	O_{min}	minimum overlap	240	480
R5	$W_{min,U}$	minimum width (U-shape)	120	240
R6	$W_{min,L}$	minimum width (L-shape)	60	120
R7	C_{min}	minimum cut spacing	80	N/A
R8	C_w	cut width	20	N/A

For each rectilinear block mask shape, Fig. 2(a) illustrates minimum width, minimum length, and minimum spacing constraints. For a given rectilinear shape, we use "length" to refer to the extent (length) of edges along the direction of metal lines, and "width" refer to the length of edges perpendicular to the direction of metal lines. When two rectilinear shapes abut each other but are not perfectly aligned, as shown in Fig. 2(b), a minimum overlap rule applies. Fig. 2(c) and (d) illustrates U-shape and L-shape constraints. Table I shows preliminary block mask rule sets ($R1$–$R8$) for 193i and 193d patterning technologies.[6]

A *selective block approach* [11] allows removal of some, but not all, segments covered by the block mask. More precisely, similar to multiple patterning technology, the selective block approach selectively removes dummy segments according to the *color* of the wire segment. There are two methodologies that realize selectivity for block mask: 1) order selectivity and 2) material selectivity. In [11], the selective blocks for metal color A and metal color B are processed sequentially. In other words, the block A for metal color A is processed right after the patterning of metal color A; then, metal B is patterned followed by block color B. Given the process order, block A only blocks metal A, and block B only blocks metal B, due to the order in which the process is assembled. By contrast, the material selectivity-based approach [9] is particularly applied to SADP/SAQP, where there are two types of wires that are created by mandrel and gap. Fig. 3 illustrates the process of the material selectivity-based approach for SAQP.

[6]We use the term "preliminary" since plan-of-record patterning strategies for mass production at N7/N5 do not yet exist. Values in Table I are from our collaborators at a leading technology development center/consortium.

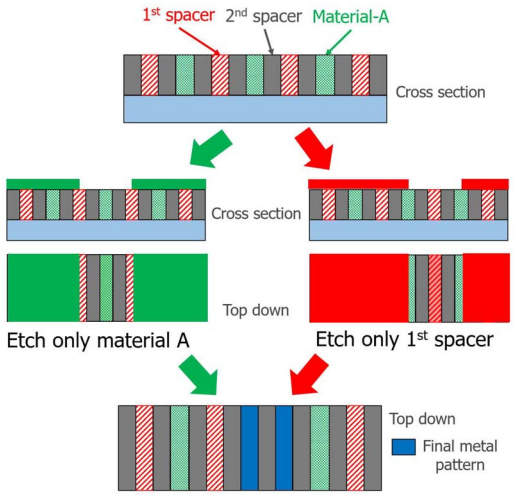


Fig. 3. Illustration of the material selectivity-based block approach.

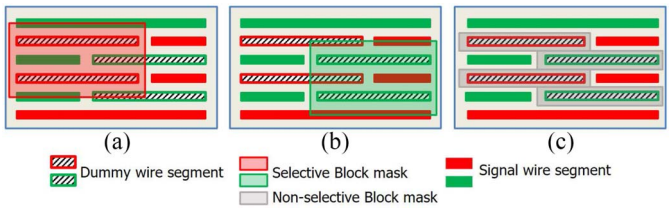


Fig. 4. Comparison between selective block and nonselective block. Selective block mask in (a) red removes only red segments, and is transparent to green segments and (b) green removes only green segments, and is transparent to red segments. (c) Complex nonselective block mask is required to remove the same dummy segments.

In this SAQP process, spacer-is-dielectric is assumed. After first and second spacers are generated, the region between spacers is filled with material A. Then, two types of block masks are introduced: one for material A, and the other for first spacers. The two block masks are used to perform the etch process which is selective to material A or to first spacer.[7] The final metal patterns are shown in blue color.

Fig. 4 illustrates the difference between the selective block and nonselective block approaches. The red (resp. green) block mask in Fig. 4(a) [resp. (b)] removes red (resp. green) dummy segments, but acts as transparent to green (resp. red) segments. Note that without selectivity, the gray block mask shape becomes complex [Fig. 4(c)] and may not be patternable with SE in 193i/193d. Since the color of wire segments is assigned alternatively track by track, selective block mask applies separately to odd and even tracks. With selectivity, as shown in Fig. 4(a) and (b), block mask shapes can extend to nontarget tracks, which is equivalent to doubling the metal pitch.

B. Cut Mask Rules/Selectivity/LELE Cut

Cut mask rules constrain shapes on the cut mask. As in [1], [3], [5], [7], and [15], we assume that cut mask shapes

[7]Indeed, there are 3 colors where each color defines the first spacer, the second spacer and the gap. However, after the second spacer formation, the first spacer is already excavated on the hardmask, and there are only the second spacer and gap as the two materials. Thus, the same color contrast that is used in SADP (e.g., two colors) can be used in SAQP as well.

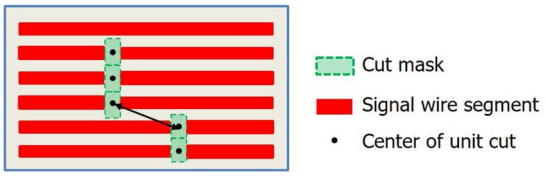


Fig. 5. Cut mask rules: minimum spacing.

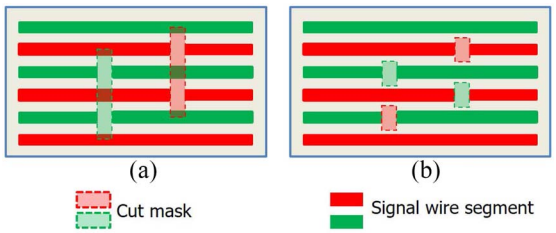


Fig. 6. Comparison between selective cuts and nonselective LELE cuts. (a) Selective cut mask in red (resp. green) realizes EOL only for red (resp. green) segments, and is transparent to green (resp. red) segments. (b) Nonselective LELE cuts may realize EOL for both colors, but a minimum cut spacing rule must be satisfied.

are unit-size rectangular cuts, with width equal to the *cut width*. A cut mask must satisfy a minimum cut spacing constraint, which is the center-to-center distance between two disjoint cuts. Two cuts are exempt from the minimum cut spacing rule if they abut and are fully aligned. For two aligned merged cuts, the minimum spacing rule is applied between each pair of unit-size cuts so that the edge-to-edge distance is always guaranteed to be above a lower bound, as shown in Fig. 5. Table I shows preliminary cut mask rule sets (*R*7 and *R*8) for 193i patterning technologies.

Similar to selective block, the *selective cut approach* realizes EOL only for the corresponding color of wire segments. As another option, the nonselective *LELE cut approach* uses two cut masks to realize EOL, regardless of the color of wire segments. Minimum cut spacing is checked within each cut mask, because two cut masks do not interfere with each other. Fig. 6(a) and (b) illustrates the selective cut and LELE cut approaches, respectively. In Fig. 6(a), similar to selective block, cuts can extend to nontarget tracks while not affecting segments of a different color. Thus, two green (resp. two red) cuts are aligned and there is no need to check minimum cut spacing since the colors of the cuts are different. Fig. 6(b) shows LELE cuts. A minimum cut spacing rule is enforced separately for two green (resp. two red) cuts.

C. Related Works

While selective block mask is a very recent concept [11], we may classify related works into four categories.

1) 1-D cut mask optimization.
2) 2-D block mask optimization.
3) 1-D cut mask-aware routing optimization.
4) 2-D block mask-aware routing optimization.

1) 1-D Cut Mask Optimization: Zhang *et al.* [16] proposed a shortest-path algorithm to resolve lithography hotspots in cut masks. Du *et al.* [3] proposed an integer linear program to minimize total EOL extension. Ding *et al.* [1] subsequently extended the methodology in [3] to reduce the runtime. Han *et al.* [7] extended the MILP formulation in [1] and proposed co-optimization of cut mask layout, dummy fill and timing. Their objective incorporates awareness of design timing in minimizing a weighted sum of EOL extensions, with weights determined by a grouping of timing slacks. Han *et al.* [7] also proposed a post-MILP optimization that iteratively removes dummy segments near timing-critical nets while satisfying density and uniformity constraints. However, 2-D block mask optimization is not supported, and the grouping-based weights that are employed to achieve a timing-aware optimization may not be accurate.

2) 2-D Block Mask Optimization: Zhang *et al.* [15] proposed a constrained shortest-path algorithm to improve the printability of 2-D block masks. Printability is assumed to be a function of the number of polygon edges in the block mask. Zhang *et al.* [15] showed a tradeoff between printability and wirelength increase, albeit without any hard design rule constraints. Ding *et al.* [1], [2] proposed an integer linear program formulation, with support of limited design rules. By contrast, our formulation supports flexible design rules, and we use recent, realistic design rules from collaborators from a leading technology consortium. We also incorporate a more accurate model to minimize timing impact of dummy fill patterns.

3) 1-D Cut Mask-Aware Routing Optimization[8]*:* Su and Chang [14] proposed a nanowire-aware router considering cut mask complexity. They first estimated the line-end probability cost for each global routing tile based on a pre-evaluation of line-end counts using minimum spanning trees. They then performed global routing while minimizing the routing bends and considering hotspots with respect to the line-end costs. After that, force-driven layer and track assignments are performed. At this stage, an attractive force is established for wires that can share a cut. Su and Chang [14] also suggested detailed routing with a cost function that considers cut sharing and EOL extension.

4) 2-D Block Mask-Aware Routing Optimization: Fang [4] proposed an ILP-based wire planning approach that considers block masks. The proposed ILP minimizes the generation of single track/wire segments during track routing. She then performed detailed routing, which is based on A* search routing with block mask-aware routing costs.

III. MILP-Based 2-D Block Mask Optimization

We now present our problem statement, our MILP formulation, as well as the timing model used in our two optimizations: 1) 2-D block mask optimization and 2) cut and block mask co-optimization.

[8]The co-optimization with routing is beyond the scope of our present work. We understand that a co-optimization of routing, cut and block mask should result in the best performance. However, integration of a custom router and a commercial tool flow with full N7/N5 design rule support is extremely hard (and, not accessible to us); "hacks" possible for us in the academic setting usually result in degraded performance.

TABLE II
NOTATIONS. THE NOTATIONS FROM THE 12TH ROW TO THE 18TH ROW (I.E., BEGINNING WITH $c^f_{i,j}$) ARE USED FOR CUT AND BLOCK CO-OPTIMIZATION

Notation	Meaning
$v_{i,j}$	(0-1) indicator of whether the block candidate j of shape i is used
$t^k_{i,j}$	delay increase due to dummy segments for net k if $v_{i,j}=0$
l_i	original dummy segment length of shape i
$r_{i,j}$	removed dummy segment length if $v_{i,j}=1$
L	total length of signal wires
K_p	set of nets in path p
$B_{q,a}(B_{q,b})$	q^{th} set of typeA (typeB) conflicting block candidates
d_{min}	minimum metal density constraint
s_p	initial timing slack of path p
m_p	timing degradation of path p
$c^f_{i,j}$	(0-1) indicator of whether cut candidate j of shape i is on cut mask f
$c_{i,j}$	(0-1) indicator of whether the cut candidate j of shape i is used
$C_{q,a}(C_{q,b})$	q^{th} set of typeA (typeB) conflicting cut candidates
j_l (j_r)	location of left (right) edge of cut or block candidate j
e_{i,x_l} (e_{i,x_r})	(0-1) indicator of whether location x is the left (right) edge of any selected cut or block candidate of shape i
e'_{i,x_l} (e'_{i,x_r})	(0-1) indicator of whether location x is the leftmost (resp. rightmost) of shape i
t'^k_{i,x_l} (t'^k_{i,x_r})	delay increase due to EOL extension for net k if $e'_{i,x_l}=1$ (resp. if $e'_{i,x_r}=1$)

A. Problem Statement

1) 2-D Block Mask Optimization: Given a post-route layout with EOL extensions and legal EOL cuts, timing information, minimum metal density constraint, and technology options (i.e., block mask rules and selectivity), perform 2-D block mask optimization considering block mask rules and metal density constraints, such that timing impact of dummy segments is minimized.

2) Cut and Block Mask Co-Optimization: Given a post-route layout, timing information, minimum metal density constraint, and technology options (i.e., cut mask rules, block mask rules, and selectivity), perform co-optimization of cut and block masks considering cut mask rules, block mask rules, and metal density constraints, such that EOL of signal segments is realized by cut or block mask, and the timing impact of EOL extension and dummy segments is minimized.

B. MILP Formulation for Block Mask Optimization

We now formulate the MILP problem for the block mask optimization problem. Table II shows notations that we use in our formulation.

1) Block Candidates: We begin by describing how a block mask layout is represented within our MILP formulation. In the block mask layout, we create a dedicated rectangular *shape* for every dummy wire segment between signal segments. Fig. 7 shows an example with three dummy wire segments, covered by three rectangular block mask shapes in the block mask layout. The final block mask layout may vary from the ones shown in Fig. 7 since each shape may change according to the selected *block candidates*. We define *block candidates* as subsegments of a rectangular block mask shape for a dummy segment. We provide several *block candidates* for each rectangular shape. We do this by slicing each rectangular shape according to a user-specified input length (120 nm, in all results reported below) into several subsegments that define block candidates.[9] Because block mask cannot realize EOL with small tip-to-tip spacing, for leftmost (or rightmost) block candidates, we add "EOL margin" between the boundary of candidates and the signal EOL. The EOL margin is illustrated in Fig. 9(a).

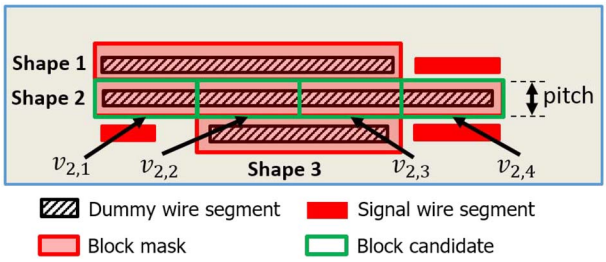


Fig. 7. Shapes and block candidates for shape 2.

Fig. 7 illustrates four block candidates $v_{2,1}$, $v_{2,2}$, $v_{2,3}$, and $v_{2,4}$ for shape 2. The block candidates are indexed in ascending (resp. descending) order of x coordinate. The final block mask layout for shape 2 is determined by selected block candidates. The height of the shape is determined by the metal pitch, as shown in Fig. 7. For the selective block approach, shapes can extend to the nontarget tracks, equivalent to doubling the metal pitch. The following MILP optimally selects block candidates of each rectangular shape, while satisfying block mask rules:

$$\text{minimize:} \quad \sum_p m_p \tag{1}$$

$$\text{subject to:} \quad \sum_{\substack{(i,j)\in B_{q,a}\\(i',j')\in B_{q,b}}} v_{i,j} + \left(1 - v_{i',j'}\right) < \left|B_{q,a}\right| + \left|B_{q,b}\right|, \ \forall q \tag{2}$$

$$L + \sum_i \left(l_i - \sum_j r_{i,j} \cdot v_{i,j} \right) \geq d_{\min} \tag{3}$$

$$\sum_{k\in K_p} \sum_{i,j} t^k_{i,j} \cdot \left(1 - v_{i,j}\right) \leq s_p + m_p, \quad \forall p \tag{4}$$

$$m_p \geq 0, \quad \forall p. \tag{5}$$

The objective is to minimize the total timing degradation arising from the final dummy fill patterns for timing-critical paths. We extract (setup) timing-critical paths using *Cadence Tempus Timing Signoff Solution v15.2* [19] (dummy segments and EOL extensions do not worsen hold, as we do not touch the clock distribution). A path is considered to be timing-critical if its slack is less than a prescribed threshold for timing-criticality.[10] For path p, the timing degradation m_p is

[9]We note that there is a tradeoff between solution quality and runtime depending on the user-specified input length, which determines fine-grained or coarse-grained block candidate generation. Experimental results for various block candidate lengths are reported in Section V.

[10]We use +200 ps as the threshold for timing-criticality in our experiments. The numbers of timing-critical paths for initial implementations are 8K, 0.9K, and 18K for M0, AES and JPEG, respectively.

Fig. 8. Illustration of a U-shape block mask rule violation.

defined as the delay increase d_p (induced by dummy fills that affect path p) that exceeds the initial timing slack s_p, i.e., $m_p = \max(d_p - s_p, 0)$.[11] In this way, we only count timing degradation that causes a negative timing slack. The value m_p is calculated from the sum of delay increases along path p, subtracted by the initial timing slack s_p.

2) Constraints for Block Mask Rule Violation: Constraint (2) prevents block mask rule violations. Given a set of close-by block candidates from neighboring shapes, we enumerate conflict sets where selection (removal) of each block candidate in any given conflict set form a violating block shape. In (2), $B_{q,a}$ (resp.$B_{q,b}$) represents conflict set q, which stores a (minimal) set of block candidates that cannot be "selected" (resp. removed) simultaneously. More specifically, we define typeA candidates such that the inclusion of the candidates forms the violating shape, and store the candidates in $B_{q,a}$. Similarly, we define typeB candidates such that the exclusion of the candidates forms the violating shape, and store the candidates in $B_{q,b}$. We create a constraint to forbid each block mask pattern that forms a block mask rule violation. Fig. 8 illustrates an example minimum width U-shape block mask rule violation on the right boundary of $v_{3,1}$. The figure shows typeA and typeB candidates that define a violating U-shape, with don't-care candidates that do not directly contribute to the formation of the U-shape violation. In this example, we prevent the U-shape rule violation with the following constraint:

$$v_{2,1} + v_{2,2} + (1 - v_{3,1}) + v_{3,2} + v_{4,2} + v_{4,3} < 6. \quad (6)$$

In (6), if any candidate in the typeA candidate set (e.g., $v_{2,1}$, $v_{2,2}$, $v_{3,2}$, $v_{4,2}$, or $v_{4,3}$) is zero or any candidate in the typeB candidate set (e.g., $v_{3,1}$) is one, the violating U-shape does not exist anymore. In this case, the constraint is automatically satisfied. We note that we only enumerate "minimal" sets of typeA and typeB candidates. For example, the inclusion of candidates $v_{1,1}$, $v_{1,2}$, and $v_{4,1}$ in addition to the typeA set above (and with the exclusion of the typeB set) forms an additional violating U-shape. However, this case is forbidden by (6). Thus, $v_{1,1}$, $v_{1,2}$, and $v_{4,1}$ are don't-care candidates. In light of this, we find that for the block mask rules that we have studied, relevant combinations will exist within very small neighborhoods of any given block candidates. Thus, the complexity of enumeration of block candidate combinations to determine sets B is in practice linear in the number of block candidates (shapes).[12]

3) Constraints for Local Minimum Metal Density: Constraint (3) enforces the local minimum metal density. We obtain the total signal wire length L from the routed layout. Variable $l_{i,j}$ is the removed dummy segment length if $v_{i,j} = 1$ for shape i. $\sum_i l_i - \sum_j r_{i,j} \cdot v_{i,j}$ calculates the total dummy wire segment length. $r_{i,j}$ is the length of block candidate $v_{i,j}$. The minimum metal density is enforced locally within each clip; this is described in Section IV-B below.

4) Constraints for Timing-Critical Paths: Constraint (4) upper-bounds the timing degradation for timing-critical paths. Variable $t^k_{i,j}$ is the delay increase for net k caused by the remaining dummy segment if $v_{i,j} = 0$. We sum up the delay increase of every stage (gate and wire) on timing-critical path p and force this sum to be smaller than $s_p + m_p$. The initial path slack s_p is calculated from a design with no dummy segments. For each timing-critical path p, $m_p = 0$ indicates that the delay increase is not larger than the initial path slack s_p and thus design WNS will not worsen[13]; otherwise, $m_p > 0$. Please note that we minimize m_p for all timing critical paths p by the objective. Constraint (5) limits m_p to be a non-negative number. We also note that (5) is necessary to optimize WNS as well as TNS. If we do not have such a constraint, the algorithm might keep removing dummy segments that are associated with "less" timing critical paths instead of focusing on the most timing critical path. For example, let us suppose that there are two paths with slacks $s_1 = 10$ and $s_2 = 0$. With (5), we optimize m_2 rather than m_1 since constraints for m_2 is tighter (i.e., the second path is more critical), and it is not necessary to optimize m_1 until the lower bound of m_1 becomes negative in (4). However, if we allow m_1 to be negative, the algorithm could tradeoff m_2 for a negative m_1 to minimize the sum of m_1 and m_2.

C. MILP Formulation for Cut and Block Mask Co-Optimization

We extend the MILP in Section III-B by providing *cut candidates*. Fig. 9 illustrates block and cut candidates with one possible final layout after cut and block mask application. Fig. 9(a) and (b) shows block candidates and cut candidates, respectively. We note that the leftmost block candidate $v_{1,1}$ is generated considering a given EOL margin to allow block mask to realize the EOL of signal wire segment. We use 10 nm as the EOL margin in our experiments. To realize the EOL of the signal wire next to the block mask, we must select at least one cut or block candidate from among the

[11] For example, if $s_p = 10$ ps and $d_p = 5$ ps, then $m_p = 0$. If $s_p = -10$ ps and $d_p = 5$ ps, then $m_p = 15$ ps. Constraints (4) and (5) enforce $m_p = \max(d_p - s_p, 0)$. We note that we do not optimize for the timing degradation within positive slacks. However, our formulation can be easily adapted by designers to preserve a given amount of positive slack (i.e., timing guardband) by decreasing s_p.

[12] In our experiments, the total runtimes of conflict lists generation for M0 and JPEG are 36 and 184 s, respectively. The number of segments (shapes) in JPEG is 257K, and the number of shapes in M0 is 63K.

[13] Here, we assume the initial "WNS" is negative. For designs with positive WNS (i.e., worst slack), we can easily shift the "zero slack" threshold to establish a guardband that preserves the worst slack of the original design (see also footnote 11 above).

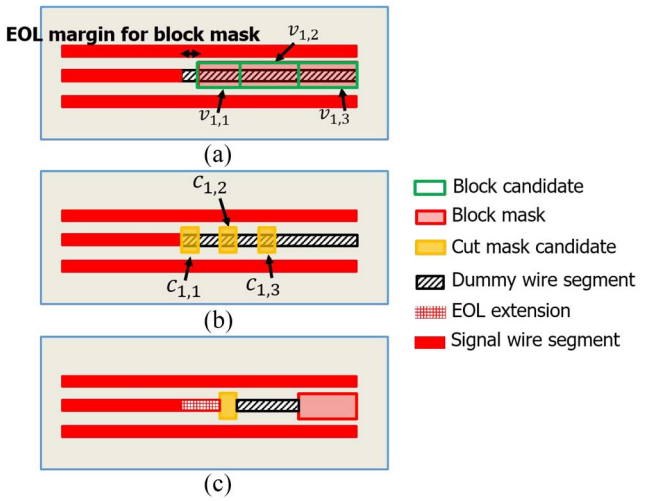


Fig. 9. Cut and block mask co-optimization. (a) Block candidates. (b) Cut candidates. (c) Possible final layout.

cut and block candidates. Fig. 9(c) shows the final layout when $v_{1,3}$ and $c_{1,2}$ are selected as the final block and cut candidate solutions, respectively.[14]

$$\text{minimize:} \quad \sum_{p} m_p \tag{7}$$

$$\text{subject to:} \quad \sum_{f} c^{f}_{i,j} = c_{i,j_l}, \ \forall i,j \tag{8}$$

$$\sum_{j} v_{i',j} + \sum_{j} c_{i',j} \geq 1, \quad \forall i' \tag{9}$$

$$\sum_{\substack{(i,j)\in B_{q,a} \\ (i',j')\in B_{q,b}}} v_{i,j} + \left(1 - v_{i',j'}\right) < \left|B_{q,a}\right| + \left|B_{q,b}\right|, \ \forall q \tag{10}$$

$$\sum_{\substack{(i,j)\in C_{q,a} \\ (i',j')\in C_{q,b}}} c_{i,j} + \left(1 - c_{i',j'}\right) < \left|C_{q,a}\right| + \left|C_{q,b}\right|, \ \forall q \tag{11}$$

$$e_{i,x_l(x_r)} \geq v_{i,j}, \ \text{if } j_l(j_r) = x, \ \forall i \tag{12}$$

$$e_{i,x_l(x_r)} \geq c_{i,j}, \ \text{if } j_l(j_r) = x, \ \forall i \tag{13}$$

$$e_{i,x_l(x_r)} \leq v_{i,j} + c_{i,j'} \ \text{if } j_l(j_r) = x, \ j'_l(j'_r) = x, \ \forall i \tag{14}$$

$$e_{i,x_l} - \sum_{x'<x} e_{i,x'_l} - e'_{i,x_l} \leq 0$$

$$e_{i,x_r} - \sum_{x'>x} e_{i,x'_r} - e'_{i,x_r} \leq 0, \ \forall i, \ \forall x \tag{15}$$

$$e'_{i,x_l} \leq e_{i,x_l}, \ e'_{i,x_r} \leq e_{i,x_r}, \ \forall i \tag{16}$$

$$\sum_{x_l} e'_{i,x_l} \leq 1, \ \sum_{x_r} e'_{i,x_r} \leq 1, \ \forall i \tag{17}$$

$$L + \sum_{i} \left(l_i - \sum_{j} r_{i,j} \cdot v_{i,j} \right) \geq d_{\min} \tag{18}$$

$$\sum_{k\in K_p} \left(\sum_{i,j} t^{k}_{i,j} \cdot \left(1 - v_{i,j}\right) + \sum_{i,x_l} t'^{k}_{i,x_l} \cdot e'_{i,x_l} + \sum_{i,x_r} t'^{k}_{i,x_r} \cdot e'_{i,x_r} \right) \leq s_p + m_p, \ \forall p \tag{19}$$

$$m_p \geq 0, \ \forall p. \tag{20}$$

We now formulate the MILP problem for the cut and block mask co-optimization. Table II shows notations that we use in our formulation. Analogous to the block mask MILP in Section III-B, the objective is to minimize the total timing degradation arising from EOL extensions and final dummy fill patterns for timing-critical paths. s_p and m_p are calculated in the same way as in Section III-B; however, for the delay increase d_p, we now consider the impact from both EOL extensions as well as the dummy fills that affect path p.

We now describe constraints in our cut and block mask co-optimization with the exception of the minimum metal density and timing constraints since these two constraints are the same as in the block mask optimization in Section III-B.

1) Constraints for LELE Cuts: In the case of nonselective LELE cuts, (8) enforces cut uniqueness. Binary variable $c^{f}_{i,j}$ indicates whether the cut candidate j for shape i on cut mask f is selected, as shown in (8). For nonselective LELE, we assume that two cut masks are available, i.e., $f = 1, 2$. For the selective cut approach, we assume only one cut mask is available, i.e., $f = 1$.

2) Constraints for EOL Realization: Constraint (9) enforces EOL realization. We use index i' to indicate a shape which is the only existing shape between any two horizontally adjacent signal segments. In other words, shape i' is a dummy shape that connects two neighboring signal segments, and must be split by cut or block to realize the EOL of the two signal segments. Thus, we enforce that at least one cut or block exists for shape i'.

3) Constraints for Cut and Block Mask Rule Violation: Constraints (10) and (11) prevent cut and block mask rule violations. Constraint (10) is the same as (2) in block mask optimization. Similar to (10) for block candidates, we enumerate sets of conflicting cut candidates and prevent them from co-existing with (11).

4) Constraints for EOL Definition: Constraints (12)–(14) find the leftmost (resp. rightmost) edge for shape i from a selected cut or block candidate, since this candidate determines EOL for the signal wire segment on its left (resp. right). Binary variable e_{i,x_l} (resp. e_{i,x_r}) indicates whether location x is the left (resp. right) edge of any selected cut or block candidates for shape i. Constraints (15)–(17) describe the methodology to find the leftmost (resp. rightmost) selected cut or block candidate. Constraints (15) and (16) ensure that $e'_{i,x_l(x_r)} = 1$ if $e_{i,x_l(x_r)} = 1$ and x is the location of leftmost (rightmost) edge for shape i. Otherwise, $e'_{i,x_l(x_r)} = 0$ is forced by checking whether e variables that are associated with x' are equal to one, where $x' < x$ ($x' > x$) for e'_{i,x_l} (e'_{i,x_r}) in (15). Fig. 10 demonstrates variable e'. In the figure, we assume that $c_{1,1} = 1$ and $v_{1,3} = 1$. e variables are computed in (21) by (12)–(14). Constraints (22)–(24) correspond to (15).

[14]We note that a block candidate cannot replace a cut candidate due to the larger EOL margin for block mask shapes. That is, cut (resp. block) candidates cannot be replaced by block (resp. cut) candidates even though they might share their locations.

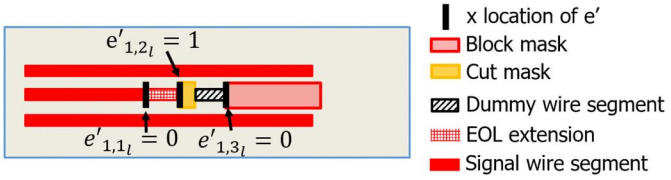


Fig. 10. Illustration of binary variable e': cut candidate $c_{1,1}$ and block candidate $v_{1,3}$ are selected.

Constraint (25) corresponds to (17). As a result, $e'_{1,2_l}$ becomes equal to one, which indicates that location $x = 2$ is the EOL, as shown in Fig. 10

$$e_{1,1_l} = 0;\ e_{1,2_l} = 1;\ e_{1,3_l} = 1 \tag{21}$$

$$e_{1,1_l} - e'_{1,1_l} \leq 0 \tag{22}$$

$$e_{1,2_l} - e_{1,1_l} - e'_{1,2_l} \leq 0 \tag{23}$$

$$e_{1,3_l} - e_{1,1_l} - e_{1,2_l} - e'_{1,3_l} \leq 0 \tag{24}$$

$$e'_{1,1_l} + e'_{1,2_l} + e'_{1,3_l} \leq 1. \tag{25}$$

TABLE III
NORMALIZED CAPACITANCE INCREASE FOR (GROUNDED) EOL EXTENSION AND (FLOATING) DUMMY FILL, USING A CADENCE INNOVUS-BASED EXTRACTION FLOW OF OUR COLLABORATORS AT A LEADING TECHNOLOGY CONSORTIUM

Case	EOL	(1)	(2)	(3)	(4)
Δcap	1270	342	53	5	1

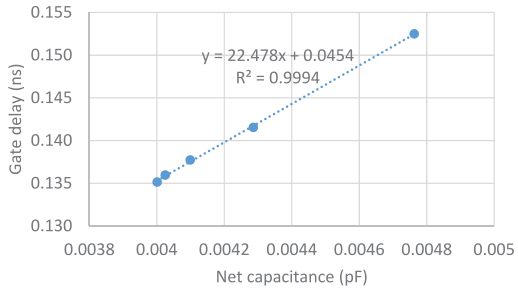


Fig. 11. Gate delay versus net capacitance for a specific gate instance.

D. Timing Model for Dummy Wire Segments

Dummy wire segments cause net capacitance increase (Δcapacitance), and hence gate and wire delay increase. This timing impact of dummy wire segments should be minimized so that the performance and robustness of designs with dummy wire segments can be consistent with (or, better than) designers' expectations at signoff. We now describe how we model Δcapacitance, along with resulting changes to gate and wire delays, to capture timing impact of dummy wire segments in our optimization flow.

1) Capacitance Model: To model the timing impact of floating dummy wire segments, we first characterize capacitance increase of signal nets due to neighboring dummy segments. Fill-aware capacitance extraction must comprehend various situations (e.g., upper/lower layers, types of neighboring wire segments of the dummy/signal wires) [6], [10]. However, to obtain linear expressions that we can incorporate into our MILP formulation, we study the impact of a dummy wire segment on capacitance of a signal wire in five simplified situations (cases) according to the distance between a signal wire and a dummy segment.

1) One track away (the dummy segment is on a neighboring track of the signal segment).
2) Two tracks away.
3) Three tracks away.
4) Four tracks away.
5) More than four tracks away.

For each case, we experiment with different parallel run lengths of the dummy wire segment to a signal wire, and measure the capacitance of the signal wire to extract the coefficients. We use *Cadence Innovus Implementation System v15.2* [17] for parasitic RC extraction with Cadence QRC [18] techfiles provided by our collaborators at a leading technology consortium.

Table III shows normalized capacitance increase per unit length for (grounded) EOL extension, and cases (1)–(4) for (floating) dummy segments from Section III-D.

2) Gate and Wire Delay Model: We use linear gate and wire delay models. The linear delay models are fast and easy to incorporate into an MILP formulation. Also, for the very small Δcapacitance values caused by dummy wire segments, linear delay modeling shows good accuracy. We use *Cadence Tempus Timing Signoff Solution v15.2* [19] to extract delays for each gate and net given extracted SPEF files of: 1) layout design with dummy wire segments for only clock nets and 2) layout design with dummy wire segments for all nets. We then use the linear delay model to extract coefficients. Timing coefficient extraction is performed for each gate instance and driving net.[15] Fig. 11 shows an example of extracted coefficients (i.e., determining a linear equation for gate delay versus capacitance) of a specific gate instance.

3) Validation of Our Timing Model: We validate our timing model by comparing with timing results obtained from *Cadence Tempus Timing Signoff Solution v15.2* [19]. We report stage and timing path delays calculated based on our model (and, which are used in our ILP formulation) and compare them with timing results from Tempus. We observe that estimated values and golden values from Tempus are close as shown in Fig. 12. The maximum errors are −4 and −23 ps (a negative value means optimistic) for stage delay and path delay, respectively. To compensate the errors, we add timing margin of 50 ps in our ILP formulation for all studies reported below.

IV. OVERALL FLOW

We now describe the overall flow of our optimizations, including conflict list enumeration and distributed optimization.

[15]We note that for different instances of the same library cell (master), the coefficients are not the same since the instances' output nets have different load capacitances according to the circuit structure. We do not separately model slew (transition time) changes that are due to the Δcapacitance changes. This is because: 1) we already achieve high accuracy by modeling each gate and net separately and 2) fill-induced slew changes are very small, since the associated capacitance and delay changes are small. Our implementation takes 20 min to extract coefficients for every gate in the *JPEG* testcase, using a single thread.

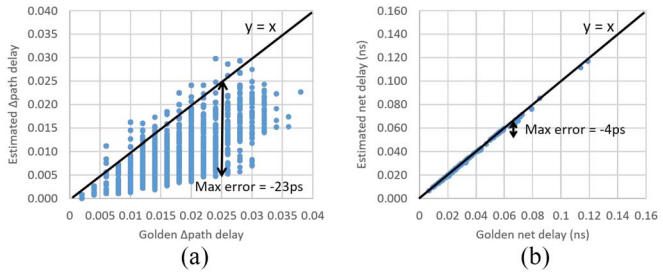


Fig. 12. Comparison of timing results from Tempus (golden) and our estimation (estimated). (a) Path delay and (b) stage delay comparisons. The maximum errors are −4 and −23 ps for stage delay and path delay, respectively.

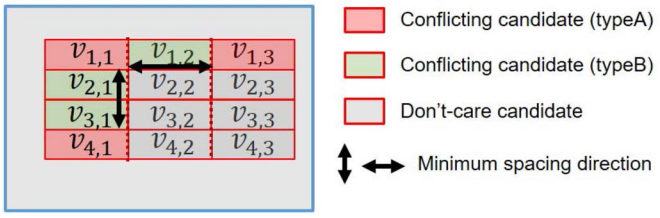


Fig. 13. Illustration of conflict list enumeration for minimum spacing constraint, showing horizontally and vertically conflicting pairs.

Algorithm 1 Enumeration for Minimum Spacing Constraint

1: **for** each block candidate pair $(v_{i,j}, v_{i',j'}) \in V$ **do**
2: **if** $S(v_{i,j}, v_{i',j'}) < S_{min}$ **then**
3: $B_{q,a} \leftarrow \{v_{i,j}, v_{i,j}\}$;
4: **for** each block candidate $v_{k,l}$ located between $v_{i,j}$ and $v_{i',j'}$ **do**
5: $B_{q,b} \leftarrow B_{q,b} \cup \{v_{k,l}\}$;
6: **end for**
7: $q \leftarrow q + 1$;
8: **end if**
9: **end for**

A. Conflict List Enumeration

1) Minimum Spacing Violation: Algorithm 1 describes the enumeration for minimum spacing constraint.

For each pair of block candidates ($v_{i,j}$,$v_{i',j'}$) within minimum spacing, we add the candidate pair to $B_{q,a}$ (line 3). They are typeA candidates, where the inclusion of each candidate (on the block mask) results in a violation (see Section III-B). We then enumerate all block candidates that are located between $v_{i,j}$ and $v_{i',j'}$ and add them to $B_{q,b}$ (lines 4–6). These candidates are typeB candidates, where the exclusion of each candidate ensures that the candidate pair ($v_{i,j}$,$v_{i',j'}$) is separated. Fig. 13 shows horizontal and vertical minimum spacing violations. For the ($v_{1,1}$,$v_{4,1}$) pair, let us assume that the vertical spacing between $v_{1,1}$ and $v_{4,1}$ is less than the minimum spacing. Then, $B_{q,a} = \{v_{1,1}, v_{4,1}\}$, and $B_{q,b} = \{v_{2,1}, v_{3,1}\}$, since $v_{2,1}$ and $v_{3,1}$ are located between $v_{1,1}$ and $v_{4,1}$. As an another example, for the ($v_{1,1}$,$v_{1,3}$) pair, let us assume that the horizontal spacing between $v_{1,1}$ and $v_{1,3}$ is less than the minimum spacing. Then, $B_{q,a} = \{v_{1,1}, v_{1,3}\}$ and $B_{q,b} = \{v_{1,2}\}$.

2) Other Design Rules: The enumeration of conflict lists for other rules can be applied similarly by collecting all typeA and typeB candidates.

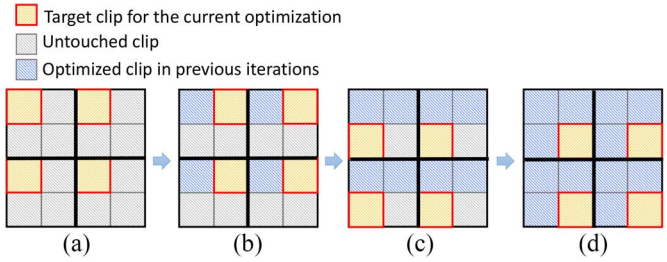


Fig. 14. Distributed optimization. (a)–(d) First, second, third, and fourth iteration in our approach. Since target clips (yellow) for an iteration do not share their boundaries with each other, each target is independently optimizable. After each iteration, the solutions in optimized clips (blue) are saved and used as boundary conditions for the next iteration for unoptimized clips (gray).

B. Distributed Optimization

The most critical limitation of the MILP-based approach in practice is runtime. To achieve a scalable approach, we adopt the distributable optimization approach that has been previously proposed by Han *et al.* [7].

We first partition the layout into small clips and optimize in four iterations. In each iteration, we select clips that are not adjacent to each other and optimize the clips in parallel. For example, we optimize all clips in the following sequence in our four iterations.

1) Clips in odd rows and odd columns in the first iteration.
2) Clips in odd rows and even columns in the second iteration.
3) Clips in even rows and odd columns in the third iteration.
4) Clips in even rows and even columns in the fourth iteration.

With this approach, as shown in Fig. 14, the target clips (yellow) do not share their boundaries with each other. Thus, each target clip can be optimized without creating any interference between clips. After each iteration, we save block/cut solutions for optimized clips. The solutions are used in the following iterations as boundary conditions.

In our implementation, we set the clip size to be 8×8 μm^2 and the boundary width to be 0.6 μm. The local minimum metal density constraint is enforced within each clip. Note that with this approach, speedup is effectively linear in compute resources. We report the results of our scalability test in Section V.

C. Overall Optimization Flow

Fig. 15 shows our overall optimization flow. We start from a routed design and candidate block (and cut) shapes that cover dummy segments. We then optimize in four iterations per metal layer. In each iteration, we optimize small clips that are independently optimizable in parallel. In an iteration, we: 1) generate block (and cut) candidates for each shape; 2) generate sets of conflict candidates with our block (and cut) mask rule checker; and 3) formulate and solve our MILP with precharacterized timing coefficients and local minimum metal density constraints. After four iterations, we obtain the optimized block/cut mask layout and perform timing/power/capacitance evaluations with *Cadence*

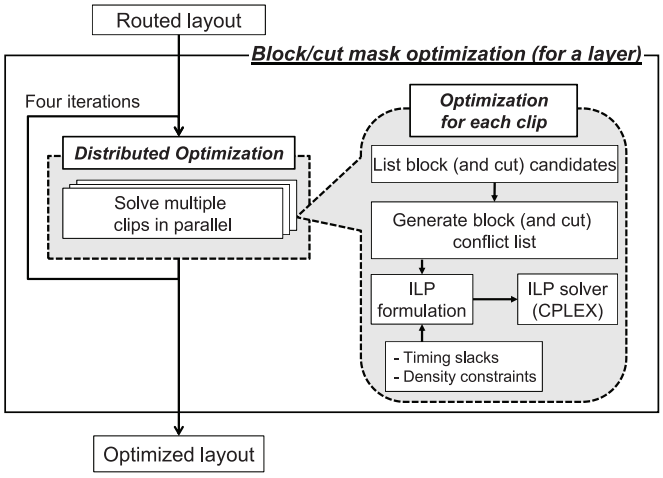


Fig. 15. Overall optimization flow.

Innovus Implementation System v15.2 [17] and *Cadence Tempus Timing Signoff Solution v15.2* [19].

V. Experimental Setup and Results

A. Experimental Setup

We implement our optimizations in C++ with *OpenAccess 2.2.6* [25] to support LEF/DEF [22], and with *CPLEX 12.5.1* [20] as our MILP solver.[16] We evaluate our approach using two design blocks (AES and JPEG) from OpenCores [23], and an ARM Cortex M0 without memories. We synthesize the designs with *Synopsys Design Compiler H-2013.03-SP3* [26] from RTL netlists and then perform placement and routing with *Cadence Innovus Implementation System v15.2* [17] using an IMEC N7 (i.e., 7-nm foundry node) library [21]. All experiments are performed with 24 threads on a 2.6-GHz Intel Xeon dual-CPU server. (As noted above, runtimes will generally see linear speedup with added compute resources.)

B. Design of Experiments

We perform three types of experiments: 1) ExptA studies the tradeoff between solution quality and runtime; 2) ExptB studies 2-D block mask optimization; and 3) ExptC studies cut and block mask co-optimization. In ExptA, we apply our optimizer to layouts with various numbers of dummy segments and clip sizes to show the tradeoff between solution quality and runtime. (We use the results to determine the best setting for input parameters.) For ExptB on 2-D block mask optimization, we use a cut mask-aware post-route layout with EOL extension already defined by a commercial tool. For ExptC on cut and block co-optimization, we perform cut and block optimization to define EOL and dummy removal using our software. We describe details of our design of experiments as follows.[17]

1) *ExptA-1:* Sensitivity study on the effect of block candidates. We tradeoff dummy removal rate and runtime for different block candidate lengths. We vary the block candidate length from 40 nm (1.2× minimum metal pitch) to 160 nm (5× minimum metal pitch) in steps of 20 nm.
2) *ExptA-2:* Sensitivity study on the effect of clip size. We tradeoff dummy removal rate and runtime for different clip sizes. We vary the clip sizes from 2 μm × 2 μm to 10 μm × 10 μm. In both experiments A-1 and A-2, we use nontiming-aware (i.e., "timing-oblivious") optimization, which is achieved by simply maximizing the removal of dummy fill.[18]
3) *ExptB-1:* Comparison of timing-aware and nontiming-aware optimizations.
4) *ExptB-2:* Comparison of the performance impact of 193i and 193d block mask rules (summarized in Table I). We use a loose 20% minimum metal density constraint to demonstrate the upper bound of performance impact from patterning technology.
5) *ExptB-3:* Comparison of the performance difference with selective and nonselective block approaches. We again use a loose 20% minimum metal density constraint to demonstrate the upper bound of performance impact from patterning technology.
6) *ExptB-4:* Comparison of the impact of metal density constraints. We study 20%, 30%, and 40% minimum metal densities.[19]
7) *ExptC-1:* Comparison of cut and block mask co-optimization to a sequential cut and block mask optimization. A cut mask only optimization is enabled without generating block shape candidates.
8) *ExptC-2:* Comparison of selective cut and LELE cut approach.

The testcases are summarized in Table IV. Table V summarizes parameter settings for each type of experiment.

C. Experimental Results

Table VII shows the experimental results of ExptB and ExptC. For ExptB, the Timing Impact Recovery column shows timing improvements. The timing impact recovery is measured in *ns* against a design with no dummy segments removed

[16]We use one thread for each CPLEX instance. Based on our experiments, solving multiple MILP instances in a serial fashion with CPLEX parallel optimization takes longer time than solving multiple MILP instances together with a single thread for each instance. For JPEG design with the same total 24 threads, the runtime with CPLEX parallel optimization is 9010 s, but the runtime with our optimization method is 4146 s.

[17]We note that it is hard to make an apples-to-apples comparison between this paper and previous works since the objectives of this paper and previous works are fundamentally different. The algorithms proposed in previous works are dedicated to solving the problem formulations posed in those works; they are difficult to extend and adapt to handle our complex design rules. For example, the work [15] simply minimizes the number of edges of each polygon of block mask patterns, and is not based on explicit design rules. Additionally, timing constraints are not considered. Similarly, the work [16] applies very limited and simple design rules, which gives a very different context from the detailed rules (obtained from our collaborators at a large industry consortium) that we use in this paper.

[18]Specifically, the nontiming-aware objective is to minimize $\sum_i (l_i - \sum_j r_{i,j} \cdot v_{i,j})$, with notations as defined in Table II. In other words, the objective is to minimize Δarea of final block mask shapes, compared to a block mask layout covering all dummy segments. Note that we disable timing-awareness by removing (4) in Section III-B.

[19]Without block mask, an SADP/SAQP-based uni-directional design implies ∼50% metal density, assuming metal width equal to spacing.

TABLE IV
TESTCASES

Expt type	Design	#Inst.	#Nets
A, B	M0	11194	11457
B	AES	10010	10066
	JPEG	52753	52778
C	M0	9884	9951
	AES	13381	13656
	JPEG	54012	54155

TABLE V
PARAMETER SETTINGS FOR THE EXPERIMENTS

ExptA				
Expt	**Timing/ nontiming**	**Layers**	**Clip width (μm)**	**Block candidate length (nm)**
A-1	nontiming	M3	2	60 - 160
A-2	nontiming	M3	2 - 10	120
Default setup	design = M0 density LB = 0% nonselective block mask			
ExptB				
Expt	**Timing/ nontiming**	**193i/ 193d**	**selective/ nonselective**	**Density LB (%)**
B-1	both	193i	selective	40
B-2	timing	both	selective	20
B-3	timing	193i	both	20
B-4	timing	193i	selective	20, 30, 40
Default setup	design = M0, AES, JPEG layers = M2, M3, M4, M5 clip size = 4 μm × 4 μm block candidate length = 120 nm			
ExptC				
Expt	**co-optimization/ sequential**		**selective/ LELE cut**	
C-1	both		selective cut	
C-2	co-optimization		both	
Default setup	design = M0, AES, JPEG layers = M2, M3, M4, M5 clip size = 4 μm × 4 μm block candidate length = 120 nm density LB = 20% 193i mask, selective block mask			

(worst case). The percentage shown indicates how closely our optimizations can approach a design that assumes all dummy segments are removed (best, or ideal, case).[20] The best and worst cases serve as extreme, baseline data points for ExptB. Table VI shows WNS, total negative slack (TNS) and switching power (P_{sw}) of the best and worst cases, At the worst case, WNS (resp. TNS) degradation is up to 0.114 ns (resp. 47.853 ns) for testcase JPEG. The switching power is increased by up to 3.4%. *Dummy removal rate* is calculated as the removed dummy segment length over the sum of removed and remaining dummy segment length.

1) ExptA-1 (Sensitivity Study on the Effect of Block Candidates): Fig. 16(a) shows dummy removal rate and runtime results for various block candidate lengths. In the range of 60–160 nm, we see that the block candidate length does not affect much dummy removal rate. However, the runtime increases proportionally to the block candidate length.

2) ExptA-2 (Sensitivity Study on the Effect of Clip Size): Fig. 16(b) shows dummy removal rate and runtime results for various clip sizes. In the range of 2–10 μm, we see that the clip size does not affect much dummy removal rate. However, the runtime increases as the clip size increases.

[20]For example, if WNS is 0.000 ns (resp. −0.100 ns) for the best (resp. worst) case, and we achieve −0.030 ns in WNS after block mask optimization, we recover 0.070 ns in WNS, with a recovery percentage of 70%.

TABLE VI
TIMING AND SWITCHING POWER OF BEST AND WORST CASES FOR EXPTA. THE UNITS ARE NS, NS, AND μW FOR WNS, TNS, AND P_{sw}, RESPECTIVELY

Design	Best case			Worst case		
	WNS	**TNS**	P_{sw}	**WNS**	**TNS**	P_{sw}
M0	-0.030	-1.737	4.06	-0.092	-23.86	4.17
AES	-0.037	-1.417	10.77	-0.069	-5.827	11.08
JPEG	-0.047	-9.583	39.18	-0.161	-57.436	40.53

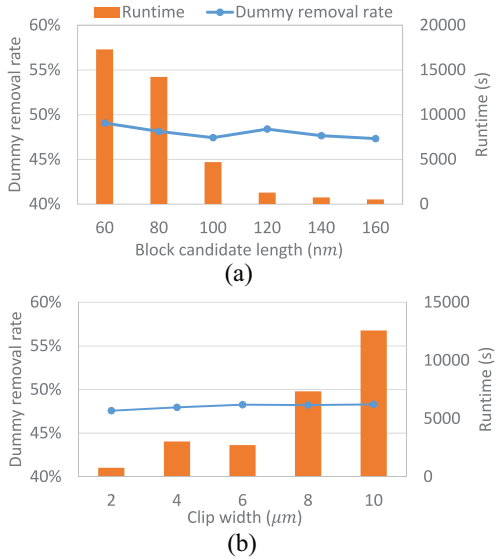


Fig. 16. Sensitivity study results. Sensitivity of the (a) block candidate length and (b) clip size on dummy removal rate.

3) ExptB-1 (Comparison of Timing-Aware and Nontiming-Aware Optimizations): We observe that nontiming-aware optimization results in higher dummy removal rates than timing-aware. However, timing-aware optimizations shows better timing impact recovery. Averaged over all three designs, timing-aware optimization recovers 57% (resp. 69%) of ΔWNS (resp. ΔTNS), compared to 32% (resp. 35%) recovered by nontiming-aware optimization. The results demonstrate that our timing-aware optimization helps recover timing with less dummy removal. We also see that the runtime of timing-aware optimization is 76% smaller on average than nontiming-aware.

4) ExptB-2 (Comparison of 193i and 193d Selective Block Mask Rules): This experiment shows the impact of patterning options. On average, application of 193i selective block mask recovers 75% (resp. 81%) of ΔWNS (resp. ΔTNS), while application of 193d selective block mask recovers 36% (resp. 48%) of ΔWNS (resp. ΔTNS). For switching power, application of 193i selective block mask recovers 53%, compared to 27% for 193d, on average. For dummy removal rate, 193i selective block mask improves by up to 43% over 193d (JPEG, metal layer M4, 62% versus 19%), with an average improvement of 21%.

5) ExptB-3 (Comparison of Selective and Nonselective Approaches): The selective block mask approach affords

TABLE VII
OVERALL EXPERIMENTAL RESULTS. VALUES IN PARENTHESES DENOTE PERCENTAGE IMPROVEMENTS (REDUCTIONS) WITH RESPECT TO THE WORST CASE AS DESCRIBED IN TABLE VI. NOTE THAT EXPTA AND EXPTB USE CUT-AWARE (FROM COMMERCIAL TOOL) AND CUT-UNAWARE POST-ROUTE LAYOUT, RESPECTIVELY. IN CONFORMANCE WITH TOOL LICENSE CONDITIONS, WE MAKE NO VALUE JUDGMENT OR COMPARISON REGARDING THE COMMERCIAL TOOL, OR BETWEEN EXPTB AND EXPTC. NO SUCH JUDGMENT OR COMPARISON IS INTENDED BY, OR TO BE INFERRED FROM, OUR REPORTED RESULTS

Experiment	Design	Option	Timing Impact Recovery (ns)		ΔP_{sw} (µW)	Dummy removal rate (%)				Runtime (s)
			ΔWNS	ΔTNS		M2	M3	M4	M5	
B-1	M0	Timing-aware	0.035 (56%)	17.307 (78%)	-0.041 (36%)	24	32	31	22	823
		Nontiming-aware	0.022 (35%)	8.945 (40%)	-0.039 (34%)	59	36	33	27	4451
	AES	Timing-aware	0.014 (43%)	2.516 (57%)	-0.129 (42%)	30	40	38	29	716
		Nontiming-aware	0.010 (31%)	1.569 (35%)	-0.116 (37%)	60	43	41	30	4231
	JPEG	Timing-aware	0.080 (70%)	34.194 (71%)	-0.458 (33%)	28	29	26	15	4150
		Nontiming-aware	0.033 (28%)	14.401 (30%)	-0.372 (27%)	56	29	27	23	11773
B-2	M0	193i	0.039 (62%)	17.681 (79%)	-0.052 (46%)	24	39	40	25	956
		193d	0.035 (56%)	13.097 (59%)	-0.034 (30%)	6	23	31	25	1963
	AES	193i	0.025 (78%)	3.482 (78%)	-0.170 (55%)	31	47	51	49	643
		193d	0.008 (25%)	1.755 (39%)	-0.095 (30%)	6	21	30	42	1307
	JPEG	193i	0.096 (84%)	40.960 (85%)	-0.759 (56%)	22	56	62	33	4146
		193d	0.030 (26%)	21.143 (44%)	-0.247 (18%)	4	16	19	10	6751
B-3	M0	Selective	0.039 (62%)	17.681 (79%)	-0.052 (46%)	24	39	40	25	956
		Nonselective	0.024 (38%)	9.280 (41%)	-0.023 (20%)	12	17	21	14	2992
	AES	Selective	0.025 (78%)	3.482 (78%)	-0.170 (55%)	31	47	51	49	643
		Nonselective	0.007 (21%)	1.414 (32%)	-0.076 (24%)	12	21	26	29	1319
	JPEG	Selective	0.096 (84%)	40.960 (85%)	-0.759 (56%)	22	56	62	33	4146
		Nonselective	0.018 (15%)	11.390 (23%)	-0.121 (8%)	7	8	10	5	6347
B-4	M0	Density LB 20%	0.039 (62%)	17.681 (79%)	-0.052 (46%)	24	39	40	25	956
		Density LB 30%	0.041 (66%)	17.577 (79%)	-0.054 (48%)	24	37	41	28	1005
		Density LB 40%	0.035 (56%)	17.307 (78%)	-0.041 (36%)	24	32	31	22	823
	AES	Density LB 20%	0.025 (78%)	3.482 (78%)	-0.170 (55%)	31	47	51	49	643
		Density LB 30%	0.025 (78%)	3.487 (79%)	-0.169 (55%)	31	46	51	49	748
		Density LB 40%	0.014 (43%)	2.516 (57%)	-0.129 (42%)	30	40	38	29	716
	JPEG	Density LB 20%	0.096 (84%)	40.960 (85%)	-0.759 (56%)	22	56	62	33	4146
		Density LB 30%	0.092 (80%)	39.868 (83%)	-0.702 (52%)	22	56	51	27	4375
		Density LB 40%	0.080 (70%)	34.194 (71%)	-0.458 (33%)	28	29	26	15	4150

Experiment	Design	Option	Timing (ns)		P_{sw} (µW)	Removal rate (%)				Runtime (s)
			WNS	TNS		M2	M3	M4	M5	
C-1	M0	Co-optimization	-0.139	-39.844	3.537	29	36	30	24	1947
		Sequential	-0.284	-136.515	3.815	12	21	18	20	1748
	AES	Co-optimization	-0.107	-21.567	18.685	29	37	35	24	1691
		Sequential	-0.132	-34.452	20.103	13	12	17	21	1460
	JPEG	Co-optimization	-0.014	-0.071	74.609	20	18	14	9	8015
		Sequential	-0.042	-0.404	70.772	9	12	12	11	8972
C-2	M0	LELE cut	-0.139	-39.844	3.537	29	36	30	24	1947
		Selective cut	-0.103	-20.482	3.475	35	37	28	20	1180
	AES	LELE cut	-0.107	-21.567	18.685	29	37	35	24	1691
		Selective cut	-0.08	-17.515	18.341	32	37	33	22	1165
	JPEG	LELE cut	-0.014	-0.071	74.609	20	18	14	9	8015
		Selective cut	-0.067	-1.293	74.669	18	18	10	7	5730

better control of dummy removal, since the minimum width of a block mask shape for a dummy segment is twice as large in the selective block mask case as in the nonselective block mask case. This results in much greater overlay margin in the selective block mask case. The results show that the selective block mask approach recovers by up to 84% and on average 75% of ΔWNS, while the nonselective block mask approach recovers up to 39% and 25% on average of ΔWNS. For ΔTNS, the selective block mask approach recovers up to 86%, and 81% on average; the nonselective block mask approach recovers up to 42% and 33% on average. Regarding ΔP_{sw}, the average recovery rates are 53% and 18% for selective and nonselective mask approaches, respectively. The timing and power benefits of the selective block approach come from high dummy removal rates; we see that the dummy removal rates are larger for the selective block approach in all designs. Fig. 17 shows layouts of M4 layer before and after dummy fill removal by optimized block masks.

6) ExptB-4 (Comparison of Different Metal Density Constraints): As metal density lower bounds increase, dummy segment removal becomes more restricted. We observe that the dummy removal rates drop by up to 36% (JPEG, M4, 51% versus 26%) with higher density constraints. With respect to timing and power, our experimental results show the expected tradeoff between timing/power and density constraints. We see that with higher density constraints, as dummy removal is more restricted, the final timing and power outcomes worsen.[21] The average percentage recovery of ΔWNS is 75% (resp. 75%, 57%) for a density lower bound of 20% (resp. 30%, 40%). The average percentage recovery of ΔTNS is 81% (resp. 81%, 69%) for a density lower bound of 20% (resp. 30%, 40%).

[21]We see that for M0, this trend is reversed between the 20% and 30% density lower bounds. The reason might be that the 20% and 30% density lower bounds are already too loose for this design so that the lower bounds do not constrain dummy removal. Similarly, we do not see much difference in timing and power for AES design.

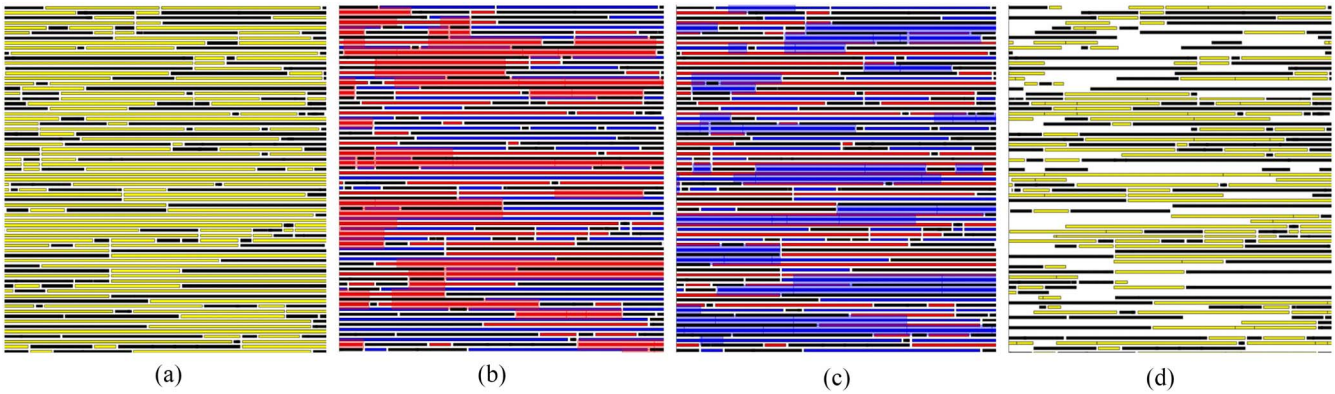


Fig. 17. Layouts of M4 layer before and after dummy fill removal by optimized block masks. (a) Initial layout with dummy fill. (b) Layout covered by the selective block mask (red). (c) Layout covered by the selective block mask (blue). (d) Layout after timing criticality-aware dummy fill removal with optimized selective block masks [notice in (d) spacing created by dummy removal around critical wire segments].

And, the recovery of P_{sw} impact is 53% (resp. 52%, 38%) on average for a density constraint of 20% (resp. 30%, 40%). For M0, we see the dummy removal rate for M4 and M5 at 20% density is slightly lower than at 30% density. The reason is that different density constraints lead to different solutions for each iteration (clip), and our timing-aware optimization does not target maximum dummy removal rate.

7) C-1 (Comparison of Co-Optimization and Sequential Optimization): We observe that WNS from co-optimization shows up to 0.146 ns improvement compared to WNS from sequential optimization. For TNS, we observe 96.671 ns (71%) improvement for M0, 12.885 ns (37%) for AES, and 0.333 ns (82%) for JPEG. We also achieve improved (reduced) switching power with our co-optimization. This is because, in the sequential approach, the EOL of all signal wire segments must be defined using only cut masks, which increases EOL extensions. On the other hand, the co-optimization approach has more flexibility with cut and block masks for the EOL realization of signal wire segments. Thus, better timing and power are achieved with smaller EOL extensions. For dummy removal rate, we also observe higher removal rate for the co-optimization, indicating that our co-optimization enables a broader solution space than the sequential cut and block approach. We emphasize to the reader that the "removal rate" for ExptC is different from "dummy removal rate" in ExptB. Removal rate is calculated as the quotient of (removed dummy segment length) divided by (sum of EOL extension length, removed dummy segment length, and remaining dummy segment length), since EOL extension is generated in ExptB.

8) C-2 (Comparison of Selective Cut Approach and LELE Cuts): Our results indicate that the selective cut approach achieves up to 36 ps better WNS compared to the LELE cut approach for M0 and AES. This is because selective cuts can be merged when they are aligned on nonadjacent tracks that are adjacent in the given color (e.g., cuts on first and third tracks) although signal segments exist in between, while LELE cuts in the same color will violate minimum spacing rule. However, for JPEG, the LELE cut approach shows better WNS. We believe that the results can be highly dependent on the routing pattern (e.g., if we have more alignment opportunity on neighboring tracks, LELE could align more cuts with the same color cut). Therefore, it is very important for the router to understand the patterning technology for the cut. Power and TNS follow the trend of WNS.

VI. Conclusion

In this paper, we first present a scalable MILP-based optimization of 2-D block masks that considers block mask rules, minimum metal density constraints, and timing impact of dummy fills. We further propose an improved timing impact model for use in our MILP formulation. A distributed optimization flow enables application of the MILP-based optimization to large design layouts. We evaluate our approach across timing-awareness, different patterning technologies, and different minimum metal density constraints. This paper shows up to 84% ΔWNS recovery and 85% ΔTNS recovery, and up to 56% Δswitching power recovery, along with up to 62% dummy removal rate. We believe that our enablement of a timing-aware optimization shows promising product-level benefits from use of 2-D block masks, and furthermore sheds light on the merits of various block mask optimization objectives. We have furthermore studied the *co-optimization* of cut and block masks. Our cut and block co-optimization opens up a broader solution space, with more flexibility in EOL realization and attendant design quality benefits. Our ongoing works include the following.

1) A more precise timing model considering interlayer coupling capacitance.
2) The co-optimization of routing, cut mask and block mask considering dummy fill impacts.
3) The optimization of block mask with clock tree-aware dummy segment removal.

References

[1] Y. Ding, C. Chu, and W.-K. Mak, "Throughput optimization for SADP and E-beam based manufacturing of 1D layout," in *Proc. DAC*, San Francisco, CA, USA, 2014, pp. 1–6.

[2] Y. Ding, C. Chu, and X. Zhou, "An efficient shift invariant rasterization algorithm for all-angle mask patterns in ILT," in *Proc. DAC*, San Francisco, CA, USA, 2015, pp. 1–6.

[3] Y. Du, H. Zhang, M. D. F. Wong, and K.-Y. Chao, "Hybrid lithography optimization with E-beam and immersion processes for 16nm 1D gridded design," in *Proc. ASP-DAC*, Sydney, NSW, Australia, 2012, pp. 707–712.

[4] S.-Y. Fang, "Cut mask optimization with wire planning in self-aligned multiple patterning full-chip routing," in *Proc. ASP-DAC*, Tokyo, Japan, 2015, pp. 396–401.

[5] W. Gillijns *et al.*, "Impact of a SADP flow on the design and process for N10/N7 metal layers," in *Proc. SPIE Adv. Lithography*, San Jose, CA, USA, 2015, Art. no. 942709.

[6] P. Gupta, A. B. Kahng, O. S. Nakagawa, and K. Samadi, "Closing the loop in interconnect analyses and optimization: CMP fill, lithography and timing," in *Proc. Int. VLSI/ULSI Multilevel Interconnect. Conf.*, Fremont, CA, USA, 2005, pp. 352–363.

[7] K. Han, A. B. Kahng, H. Lee, and L. Wang, "ILP-based co-optimization of cut-mask layout, dummy fill and timing for sub-14nm BEOL technology," in *Proc. SPIE Photomask Technol.*, Art. no. 96350E, 2015, pp. 1–14.

[8] K. Han, A. B. Kahng, and H. Lee, "Evaluation of BEOL design rule impacts using an optimal ILP-based detailed router," in *Proc. DAC*, San Francisco, CA, USA, 2015, pp. 1–6.

[9] T. Han, H. Liu, and Y. Chen, "A paradigm shift in patterning foundation from frequency multiplication to edge-placement accuracy: A novel processing solution by selective etching and alternating-material self-aligned multiple patterning," in *Proc. SPIE Alternative Lithographic Technol.*, 2016, Art. no. 977718.

[10] A. B. Kahng and R. O. Topaloglu, "A DOE set for normalization-based extraction of fill impact on capacitances," in *Proc. ISQED*, San Jose, CA, USA, 2007, pp. 467–474.

[11] C. Y. Lee, C.-Y. Ting, and J.-H. Shieh, "Method of patterning for a semiconductor device," U.S. Patent 8 697 537, Apr. 15, 2014.

[12] P. Raghavan *et al.*, "Holisitic device exploration for 7nm node," in *Proc. CICC*, San Jose, CA, USA, 2015, pp. 1–5.

[13] S. M. Y. Sherazi *et al.*, "Architectural strategies in standard-cell design for the 7 nm and beyond technology node," *J. Micro/Nanolithography MEMS MOEMS*, vol. 15, no. 1, pp. 1–11, 2016.

[14] Y.-H. Su and Y.-W. Chang, "Nanowire-aware routing considering high cut mask complexity," in *Proc. DAC*, San Francisco, CA, USA, 2015, pp. 1–6.

[15] H. Zhang, Y. Du, M. D. F. Wong, and K.-Y. Chao, "Mask cost reduction with circuit performance consideration for self-aligned double patterning," in *Proc. ASP-DAC*, Yokohama, Japan, 2011, pp. 787–792.

[16] H. Zhang, Y. Du, M. D. F. Wong, and K.-Y. Chao, "Lithography-aware layout modification considering performance impact," in *Proc. ISQED*, Santa Clara, CA, USA, 2011, pp. 1–5.

[17] *Cadence Innovus Implementation System*. Accessed on Mar. 29, 2017. [Online]. Available: https://www.cadence.com/content/cadence-www/global/en_US/home/tools/digital-design-and-signoff/hierarchical-design-and-floorplanning/innovus-implementation-system.html

[18] *Cadence Quantus QRC Extraction Solution*. Accessed on Mar. 29, 2017. [Online]. Available: https://www.cadence.com/content/cadence-www/global/en_US/home/tools/digital-design-and-signoff/silicon-signoff/quantus-qrc-extraction-solution.html

[19] *Cadence Tempus Timing Signoff Solution*. Accessed on Mar. 29, 2017. [Online]. Available: https://www.cadence.com/content/cadence-www/global/en_US/home/tools/digital-design-and-signoff/silicon-signoff/tempus-timing-signoff-solution.html

[20] *IBM ILOG CPLEX*. Accessed on Mar. 29, 2017. [Online]. Available: http://www.ilog.com/products/cplex/

[21] *IMEC*. Accessed on Mar. 29, 2017. [Online]. Available: http://www.imec-int.com/en/home

[22] *LEF/DEF 5.7 Reference*. Accessed on Mar. 29, 2017. [Online]. Available: http://projects.si2.org/openeda.si2.org/projects/lefdef

[23] *OpenCores: Open Source IP-Cores*. Accessed on Mar. 29, 2017. [Online]. Available: http://www.opencores.org

[24] *OpenMP*. Accessed on Mar. 29, 2017. [Online]. Available: http://www.openmp.org/

[25] *Si2 OpenAccess*. Accessed on Mar. 29, 2017. [Online]. Available: http://projects.si2.org/?page=69

[26] *Synopsys Design Compiler*. Accessed on Mar. 29, 2017. [Online]. Available: http://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/dc-ultra.html

Peter Debacker (S'04) received the M.Sc. degree in electrical engineering (with high distinction) from KU Leuven, Leuven, Belgium, in 2004.

He was with Philips, Amsterdam, The Netherlands, researching on intelligent control algorithms for digital video projection and implemented ultralow-power ISM transceivers with Essensium, Leuven. He joined IMEC, Leuven, in 2011, researching on energy efficient high throughput digital baseband SoCs, where he is currently a Senior Researcher with the Process Technology Department. His current research interests include low-power digital chip and processor architectures and implementation in advanced technology nodes.

Kwangsoo Han (S'11) received the B.S. and M.S. degrees in electrical engineering from Hanyang University, Seoul, South Korea. He is currently pursuing the Ph.D. degree with the University of California at San Diego, La Jolla, CA, USA.

His current research interests include design for manufacturability and very large-scale integration physical design optimization.

Andrew B. Kahng (M'03–SM'07–F'10) received the Ph.D. degree in computer science from the University of California at San Diego, La Jolla, CA, USA.

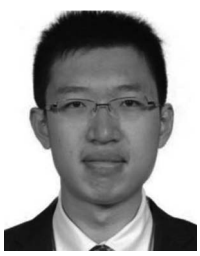
He is a Professor with the Computer Science Engineering Department and the Electrical and Computer Engineering Department, University of California at San Diego. His current research interests include IC physical design, the design-manufacturing interface, combinatorial optimization, and technology roadmapping.

Hyein Lee (S'12) received the B.S. degree from Yonsei University, Seoul, South Korea, and the M.S. degree from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2009, both in electrical engineering. She is currently pursuing the Ph.D. degree with the VLSI CAD Laboratory, University of California at San Diego, La Jolla, CA, USA.

From 2009 to 2012, she was with the Design Technology Team, Samsung Electronics, Suwon, South Korea. Her current research interests include low-power design optimization and design-manufacturing interface.

Praveen Raghavan (M'04) received the B.S. degree in electrical engineering from REC Tiruchirappalli, Tiruchirappalli, India, the M.S. degree from Arizona State University, Tempe, AZ, USA, and the Ph.D. degree from KU Leuven, Leuven, Belgium, in 2009.

In 2007, he was a Visiting Researcher with the Berkeley Wireless Research Center, University of California at Berkeley, Berkeley, CA, USA. He is currently the Distinguished Engineer in IMEC leading the group for design-enabled technology exploration for device-design co-optimization on CMOS and beyond CMOS technologies.



Lutong Wang (S'16) received the B.S. degree in microelectronics from Tsinghua University, Beijing, China, in 2014, and the M.S. degree in electrical and computer engineering from the University of California at San Diego, La Jolla, CA, USA, in 2016, where he is currently pursuing the Ph.D. degree.

His current research interests include physical design implementation and DFM methodologies.