Auxiliary pattern-based optical proximity correction for better printability, timing, and leakage control

Andrew B. Kahng
University of California at San Diego
Computer Systems Engineering Department
and
Electrical and Computer Engineering Department
La Jolla, California 92093

Swamy Muddu
Chul-Hong Park
University of California at San Diego
Computer Systems Engineering Department
La Jolla, California 92093
E-mail: chpark@vlsicad.ucsd.edu

Abstract. Optical proximity correction (OPC) is a mandatory resolution enhancement technique (RET) to ensure the printability of layout features in silicon. The most prominent OPC method, *model-based OPC*, alters the layout data for the photomask that enables drawn layout features to be accurately reproduced by lithography and etch processes onto the wafer. This technique in various forms has now become standard in integrated circuit (IC) manufacturing at 0.18 μm and below. However, model-based OPC is computationally expensive and its runtime increases with technology scaling. The *cell-based OPC* approach improves runtime by performing OPC once per cell *definition*, as opposed to once per cell *instantiation* in the layout. However, cell-based OPC does not comprehend intercell optical interactions that affect feature printability in a layout context. This leads to printability, and consequently, performance and leakage, degradation. In this work, we propose *auxiliary pattern-enabled* cell-based OPC to improve printability of cell-based OPC, while retaining its runtime advantage. Auxiliary patterns (AP) are nonfunctional poly features that are added around a standard cell to "shield" it from optical proximity effects. We present the AP-based OPC approach and demonstrate its advantages over cell-based and model-based OPC in terms of printability as well as timing and leakage variabilities. AP-based OPC improves the edge placement error over cell-based OPC by 68%. To enable effective insertion of AP in cell instances at a full-chip layout level, we propose a dynamic programming (DP)-based method for perturbation of detailed placement. Our approach modifies the detailed placement to allow opportunistic insertion of AP around cell instances in the design layout. By perturbing placement, we achieve 100% AP applicability in designs with placement utilization less than 70%. AP-based OPC also reduces leakage and timing variability compared to conventional cell-based OPC. We further demonstrate that AP insertion achieves timing and leakage variability comparable to that of model-based OPC.
[DOI: 10.1117/1.2898504]

Subject terms: OPC; cell-based OPC; model-based OPC; auxiliary patterns; performance; EPE.

Paper 07046R received Aug. 7, 2007; revised manuscript received Sep. 25, 2007; accepted for publication Oct. 30, 2007; published online Apr. 2, 2008. This paper is a revision of a paper presented at the SPIE conference on Photomask Technology 2006, September 2006, Monterey, California. The paper presented there appears (Ref. 13) in SPIE Proceedings Vol. 6349.

1 Introduction

Optical proximity correction (OPC) is a key resolution enhancement technique (RET) that enables fabrication of integrated circuit (IC) features using subwavelength optical lithography. OPC modifies the shapes of IC layout features to enable their printability in silicon. In sub-180-nm technology nodes, OPC is performed by iterative modification of layout feature edges. The iterative correction is performed until the resulting simulated image matches the target layout. The correction process itself can be driven by simulation using models of lithography and wafer processing steps during fabrication. Specifically, the models used for OPC describe the relationship between pattern information and aerial image, resist, and etch process parameters.[1] However, this approach, which we refer to as model-based OPC (MBOPC), is computationally expensive (because of its iterative nature). Since MBOPC relies on simulation, its runtime has grown unacceptably with each successive technology generation, and it has emerged as a major bottleneck for turnaround time (TAT) of IC data preparation and manufacturing.

To address the OPC runtime issue, a cell-based OPC (COPC) approach has been proposed in Refs. 2 and 3. The COPC approach runs OPC once per each cell definition (i.e., per "cell *master*") rather than once per placement or unique instantiation of each cell (i.e., per "cell *instance*"). In other words, in the COPC approach, master cell layouts

1537-1646/2008/$25.00 © 2008 SPIE

in the standard-cell library are corrected before the placement step, and then placement and routing steps of IC design flow are completed with the corrected master cells. Since COPC is performed *once* for all cell masters in the library, it achieves significant OPC runtime reduction over MBOPC, which is performed at the full-chip layout level for every design that uses the cells. Unfortunately, optical proximity effects (OPEs) in lithography cause interaction between layout pattern geometries. Since the neighboring environment of a cell in a full-chip layout is completely different from the environment of an isolated cell, the COPC solution can be incorrect when instantiated in a full-chip layout. As a result, there can be a significant discrepancy in printed feature critical dimension (CD) between COPC and MBOPC solutions.

In this work, we devise a novel *auxiliary pattern* (AP) technique (Ref. 13) that *shields* poly patterns near the cell outline from the proximity effect of neighboring cells. Consequently, COPC with AP achieves the same printability as MBOPC, but without any runtime overhead. APs inserted at the cell boundary reduce the difference between OPC impact of a cell in an isolated and layout context. This effectively allows the *substitution* of an OPC'ed cell with APs directly in the layout. Auxiliary patterns are vertical (V-AP) and/or horizontal (H-AP) nonfunctional (dummy) poly lines. *V-AP features are located within the same cell row and print on the wafer. H-AP features are located in the overlap region between cell rows; their width is comparable to that of subresolution assist features (SRAFs) and they do not print on the wafer.*

Optimization of RET by elongating design features and adding trim to SRAF features was recently proposed by Wallace and Jang.[4] In this technique, line ends facing a gap are elongated, and SRAFs are added between them. To reduce the gap between SRAF and the features, the SRAF patterns are trimmed and included on the photomask. This increases the contrast in the line end, thereby improving line-end shortening. This technique does not have any layout area impact, but it does not improve SRAF continuity at the boundary between different cells. Garg et al.[5] recently proposed a technique for insertion of dummy poly lines in empty spaces between poly gates within cell layouts. The dummy poly features are added as extensions to existing poly lines. The insertion of dummy poly improves the regularity of poly and enables tuning of the OPC recipe for improved process windows. However, this technique has a 5 to 11% cell area impact, which can translate to a design level area increase.

In contrast to the SRAF-based and dummy-poly-based approaches presented in recent literature, our approach seeks to minimize the difference between cell-based OPC and model-based OPC solutions by inserting dummy poly lines (auxiliary patterns, or APs) on all sides of a cell instance. To facilitate insertion of AP for some cell instances in the design placement (i.e., layout), it is helpful to perturb cell locations for some types of AP, as detailed in Sec. 3.3. *Indeed, to maximize the total amount of AP insertion in all cells in the design, we perturb detailed placement of standard cells using a dynamic programming (DP)-based approach*. This allows opportunistic instantiation of AP around cell instances, depending on the availability of free space in the layout. Note that placement perturbation does not increase the design area; it merely readjusts cell locations amidst the white space to allow AP insertion. We achieve 100% AP insertability in placements with row utilization less than 70%. In designs with row utilizations of 80 and 90%, the insertability of AP decreases to 98 and 80%, respectively, (due to lack of white space for AP insertion in cells). The movement of cells in the detailed placement may result in potential design level timing impact. To minimize the impact of the dynamic programming-based approach on design timing, we perform timing-aware modifications of cell placement. In our approach, we do not perturb the locations of timing critical cells nor, consequently, the routes connected to them.

Apart from runtime improvement, AP-based OPC can be used to enhance the accuracy of postlitho timing and leakage analysis. Lithography simulation-based design analysis, optimization, and sign-off is becoming a necessity in the sub-100-nm technology nodes.[6] However, performing chip-level lithography simulation is computationally expensive. Furthermore, two instances of the same standard cell will print differently based on their respective placement neighborhoods. This necessitates the creation of multiple variants for each cell to perform postlitho timing and leakage analysis. Ideally, it is preferable to use a *single* aerial image of every standard cell for postlitho analysis at a specific process condition. AP-based OPC allows this by shielding cells from their neighbors. Cao, Dobre, and Hu[7] recently proposed a methodology for standard-cell characterization considering litho-induced systematic variations. The objective of their work is to enable efficient postlitho analysis by running litho-aware characterization. To minimize the difference between the isolated and placement context of a standard cell, vertical dummy poly patterns are inserted at the cell boundary. Our approach differs from that of Ref. 7 in two main aspects: 1. we perform dummy poly insertion on *all sides of a cell* to shield OPE, and 2. we perform opportunistic, timing-aware insertion of AP at the *full-chip level by perturbing detailed placement*. In other words, we use detailed placement to maximize the insertion of AP in cell instantiations.

In our approach, vertical (V)-APs are designed to print on the wafer to shield the proximity effectively. V-AP width and spacing can be adjusted depending on the extent of OPE at the placement level. We also insert nonprinting H-APs to shield OPE between cell rows. Placement of H-APs reduces line-end pullback, resulting in improvement of the curvature of poly litho contours. The curvature of poly around a line end extends deep into the device region (i.e., poly over diffusion), illustrated in Fig. 1. A decrease in the extent of line-end pullback improves performance and leakage variability.

The primary objective of our work is to reduce OPC turnaround time without any impact on timing and leakage of the design. Our main contributions are as follows.

- We propose a novel approach for application of COPC to designs, based on the insertion of APs. APs minimize CD difference between COPC and MBOPC. Consequently, AP-enabled COPC achieves significant reduction in runtime compared to MBOPC. We demonstrate that COPC with V-AP and a combination of

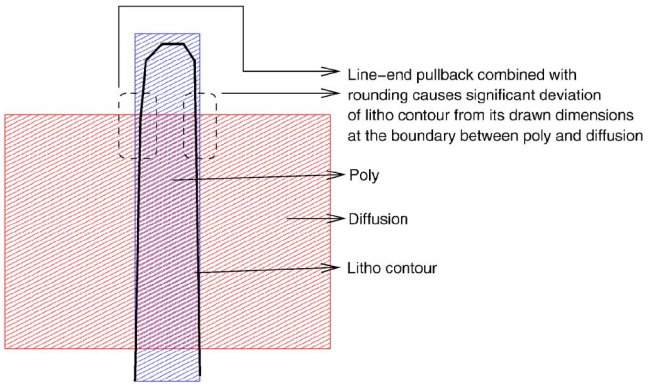

Fig. 1 Line-end pullback combined with rounding can impact device corner line width significantly.

V- and H-AP can achieve better edge placement error (EPE) than conventional COPC.

- AP insertion might not be feasible on all instantiations of a standard cell in the design. To enable AP insertion in all cell instantiations in the layout with no area penalty, *we propose an opportunistic, DP-based methodology for perturbation of detailed placement to allow AP insertion*. The perturbation of placement maximizes the opportunity for AP insertion. However, detailed placement changes can potentially lead to change in design timing. We minimize the timing impact by introducing timing awareness in our DP-based perturbation approach. All cell instances on critical paths are marked as fixed and are not moved during placement perturbations. This ensures that all routes connected to these critical instances (and their cell delays) do not change during subsequent engineering change order (ECO) routing steps. (ECO steps are executed by the place-and-route tool to perform minor modifications to design layout.)
- Using a litho-aware characterization methodology, we demonstrate an average improvement of 65 and 42%, respectively, in leakage and timing variability of AP-based OPC. At the cell layout level, we also show that timing and leakage behavior of AP-based OPC is comparable to that of MBOPC. Since full-chip analysis of postlitho timing and leakage power is not feasible, we compare EPE of all poly features between AP-based OPC and MBOPC. We show that AP-based OPC achieves EPE comparable to that of MBOPC at the full-chip level.

This work is organized as follows. We evaluate CD impact of AP in terms of line width, line end, and contact poly in Sec. 2. In Sec. 3, we discuss AP generation, printability impact, and a placement perturbation method for improving the feasibility of AP insertion. In Sec. 4, we discuss details of the litho-aware timing and leakage characterization flow. We use the flow to demonstrate improvement in timing and leakage variability of AP-based OPC over COPC. We discuss our experimental setup and results in Sec. 5. In Sec. 6, we summarize our contributions.

2 Critical Dimension Impact of Auxiliary Pattern

The key role of the auxiliary pattern technique is to shield poly patterns near the cell outline from proximity effects of neighboring cells. We devise three test structures to evaluate the CD impact of AP in terms of line width, line end, and contact poly (Contract poly defines the overlapped area of poly and contact, which may also be called "contract coverage."). Each test structure has two test cells that consist of line width of 0.1 μm, pitch of 0.3 μm, and line length of 2.0 μm. For simulation of CD impact, vector aerial image simulation is performed with wavelength $\lambda=193$ nm and NA=0.7 for 90 nm. Annular illumination with $\sigma=0.85/0.57$ is used. For scattering bar (SB) insertion rules, SB width=0.04 μm, SB-to-poly spacing=0.12 μm, and SB-to-SB spacing=0.12 μm are used.

Figure 2(a) shows a test pattern structure to evaluate the CD impact of AP on line width. We use AP width of 0.1 μm, AP-to-poly space of 0.13 μm, and AP-to-AP space of 0.14 μm. AP can be inserted as long as the space between the border poly is greater than 0.36 μm. This spacing is determined by the minimum design rule. Figure 2(b) shows the CD impact of AP on line width. In this plot, the x axis indicates the space between border poly of two adjacent cells, and the y axis indicates the CD difference between MBOPC and COPC, measured in terms of CD. The

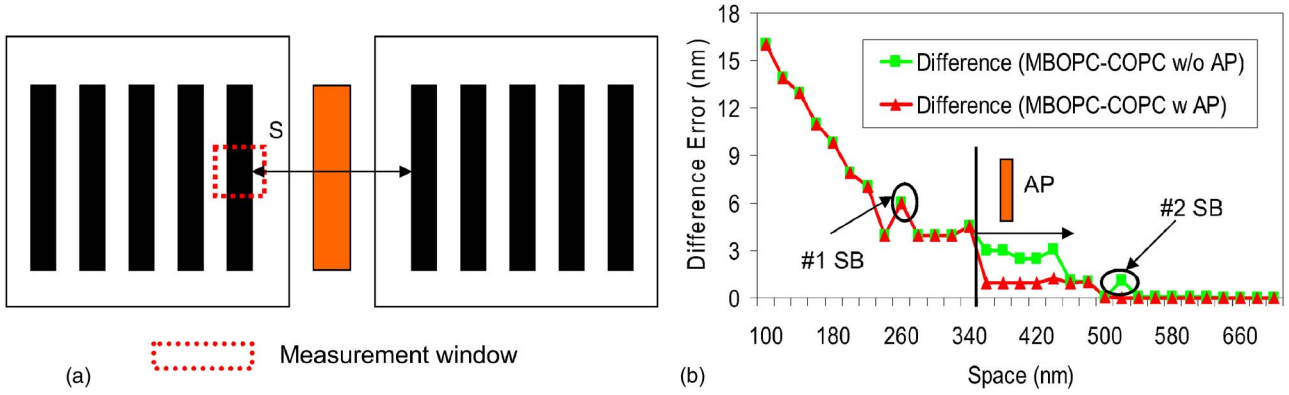

Fig. 2 CD impact of AP on line width: maximum CD differences between COPC and MBOPC are 3 nm without AP and line 1 nm with AP. The width of vertical-AP is as large as the minimum line width of a feature on the poly layer.

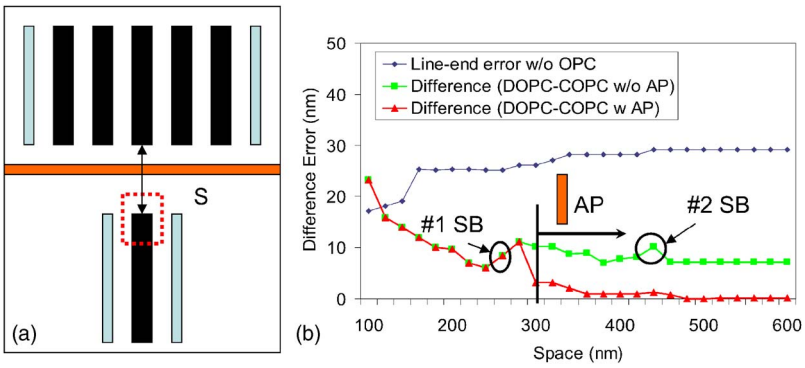

Fig. 3 CD impact of AP at line end: maximum CD differences between COPC and MBOPC are 10 nm without AP and 3 nm with AP. The width of horizontal-AP is as small as that of a subresolution assist feature (SRAF), since there is an active layer at the boundary between different cell rows.

maximum CD difference between MBOPC and COPC without AP is 3 nm, while the maximum difference with AP is only 1 nm.

The proximity shield effect of AP with respect to line-end shortening is shown in Fig. 3(b). We use a horizontal AP width of 0.4 μm for proximity shielding. The minimum space between line-end poly for insertion of APs is 0.3 μm. The maximum CD difference between MBOPC and COPC without AP is 10 nm, while the maximum difference with AP is about 3 nm. The CD difference thus is reduced by up to 75% with AP insertion. We also evaluate the effectiveness of AP with respect to contact poly, which is the closest geometry to neighboring cells. The minimum space between contact poly for insertion of AP is 0.36 μm, as shown in Fig. 4(b). The maximum CD difference between MBOPC and COPC without AP is 5 nm, while the maximum difference with AP is about 1.5 nm. Consequently, COPC with AP achieves the same printability as MBOPC with respect to line patterning issues.

3 Auxiliary Pattern Methodology

In this section, we discuss details of AP generation, placement perturbation, and a modified design flow to enable AP-based OPC.

3.1 *Auxiliary Pattern Generation*

Auxiliary patterns overcome the deficiencies of the COPC approach for standard-cell layouts. AP features consist of vertical (V-AP) and/or horizontal (H-AP) dummy poly as shown in Figs. 5(a) and 5(b). V-AP features are located within the same cell row as the standard cell, while H-AP features are located in the overlap region between adjacent cell rows. Devices in the layout are typically laid out vertically (assuming horizontal cell rows). Since the impact of lithography on gate CD is more interesting from a designer's perspective, patterns laid out vertically at cell boundaries within the same cell row should be shielded from proximity effects for maximum value and accuracy of cell-based OPC. Thus, the width of V-AP is as large as the minimum linewidth of a feature on the poly layer. On the other hand, the width of H-AP is as small as the width of a subresolution assist feature (SRAF). H-AP differs from the SRAF technique in that the location of SRAFs depends on the distance between poly lines, while the AP is located exactly at the cell boundary. In general, there is an active layer at the boundary between different cell rows, and hence the H-AP must not be allowed to print on the wafer. There are three types of V-AP according to the location of insertion. We now describe the three types of V-AP as follows.

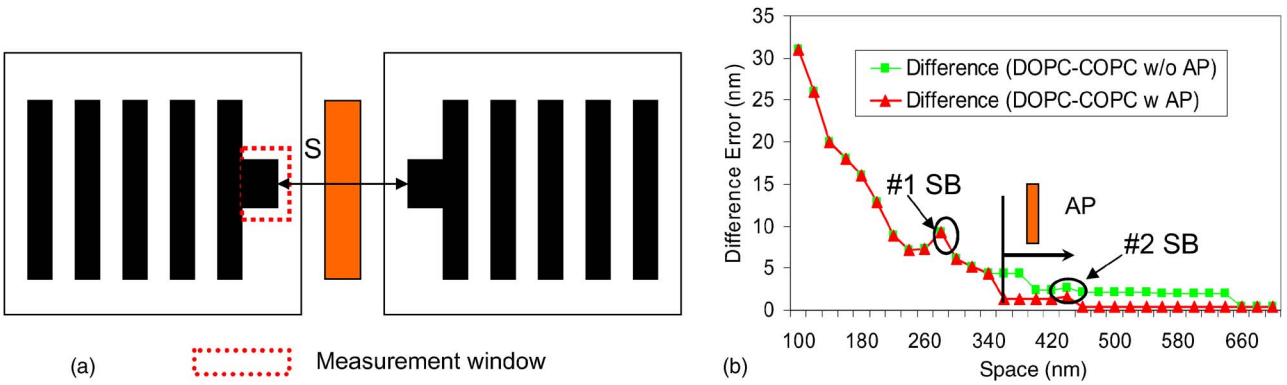

Fig. 4 CD impact of AP in contact poly: maximum CD differences between COPC and MBOPC are 5 nm without AP and 1 nm with AP.

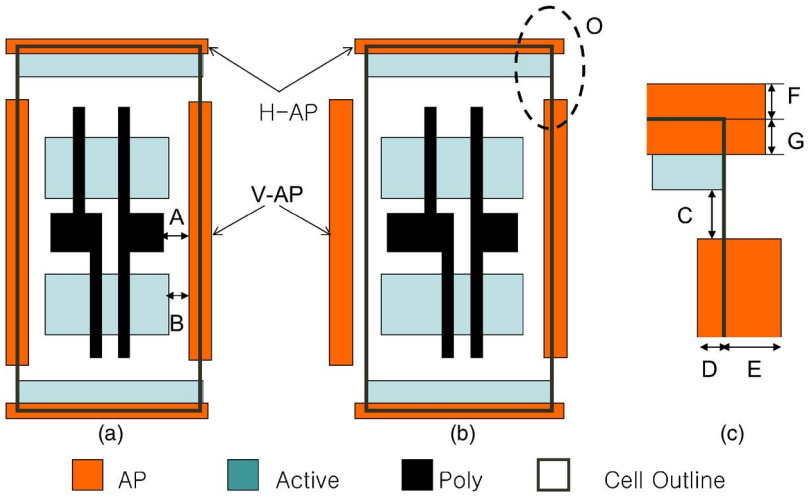

Fig. 5 Examples of standard-cell layouts with APs: (a) type-1 V-AP, (b) type-2 V-AP, and (c) is an enlargement of the region O of (b).

3.1.1 *Type-1 vertical auxiliary pattern*

Figure 5(a) illustrates a type-1 V-AP located at the center of (i.e., centered about) the cell outline, such that the left width [D in Fig. 5(c)] is the same as the right width of cell outline to right edge of V-AP [E in Fig. 5(c)]. Spaces A and B, respectively, define the space between border poly and AP, and the space between active-layer geometry and V-AP. Rule A typically means the minimum design rule of poly-to-poly space. However, to insert at least one SRAF between border poly and AP, rule A can be the poly-to-poly spacing for inserting one SRAF. Since A and B in a typical standard cell are smaller than the required minimum spacing, it is desirable for the pattern geometries of each standard cell to be modifiable to permit the instantiation of cells with a type-1 V-AP.

3.1.2 *Type-2 vertical auxiliary pattern*

Type-2 V-AP locations satisfy both A and B of minimum design rules, as shown in Fig. 5(b). Width D is different from width E. The type-2 V-AP can also be placed outside the cell outline. In Fig. 5(c), which is an enlargement of the region O of Fig. 5(b), C is the space between V-AP and the active layer, and is the same as the minimum space between the poly line end and the active layer. The width from cell outline to the bottom edge G of the H-AP is the same as the width between cell outline and the top edge F of AP.

3.1.3 *Type-3 vertical auxiliary pattern*

Figure 6 illustrates a type-3 V-AP that is placed at the center of the placement site. Since placing the type-3 V-AP at the center of the site achieves enough space between poly and AP, the type-3 V-AP can maintain minimum space rules such as poly-to-poly and poly-to-active spacing while simultaneously minimizing the area penalty.

Various auxiliary patterns can be constructed by combinations of the prior three types of APs. Figure 7 shows two examples: 1. a two-cell placement with a combination of cells with type-1 and type-2 APs; and 2. a two-cell placement with a combination of cells with type-1 and type-3 V-APs. Thus, in the application of the AP technique, all combinations of all possible types of AP are feasible and can be considered. In addition, Figs. 7(a) and 7(b) show APs completely overlapped or having certain required spacing to each other, respectively.

3.2 *Area Penalty with Auxiliary Pattern*

In this section, we discuss the area impact of AP insertion and its trend with technology scaling (i.e., design rule shrinkage). Standard cells with AP can increase cell area in the layout. For type-1 V-AP, APs are located at the center of the cell outline, and hence, the area penalty is equal to the width of AP. In the case of type-2 V-AP, the area penalty in a cell depends on the spacing between border poly and cell outline, and the spacing between the border active layer geometry and the cell outline. In this case, the penalty is equal to 2× the sum of the AP width and the spacing to satisfy both A and B of minimum design rules. The layout of type-3 V-AP depends on the width of the placement site. The penalty with type-3 V-AP is the sum of the placement site width and the AP width.

The proximity shield effect of AP is affected by the shrinkage of complementary metal-oxide semiconductor (CMOS) design rules. The decrease of feature pitch due to technology scaling affects the optical proximity between

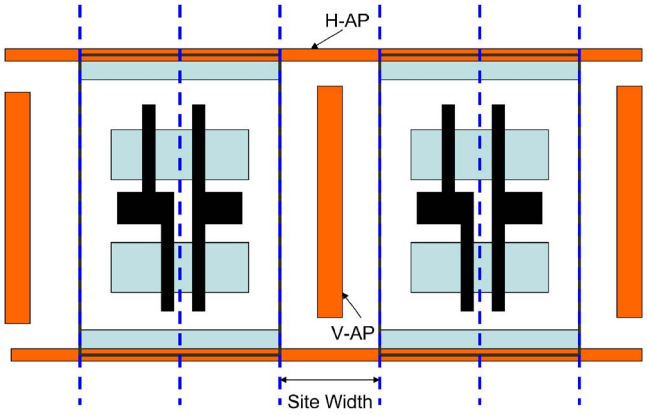

Fig. 6 An example of a standard-cell layout with type-3 V-AP.

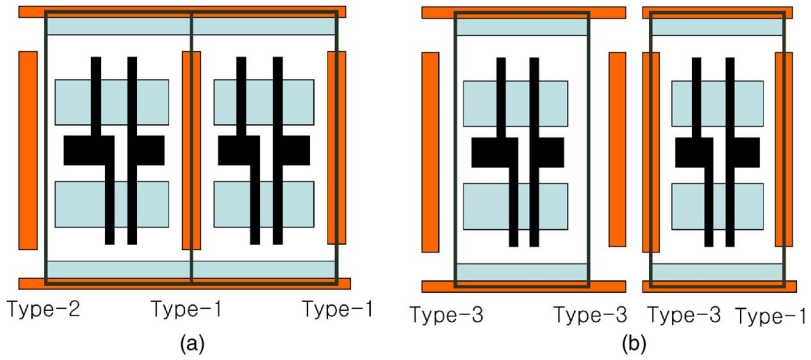

Fig. 7 Standard cell layouts constructed by combinations of the three types of APs: (a) a two-cell layout with type-1 and type-2 V-APs and (b) a two-cell layout with a combination of type-1 and type-3 V-APs.

layout features. This has implications for OPC and consequently AP insertion. An increase in the number of features within a fixed optical interaction region results in an increase in the proximity effects between them. This may necessitate insertion of an increased number of AP at cell boundaries to shield from the proximity of neighbors. However, the optical interaction radius scales with technology. The optical proximity range (OPR) depends on the optical wavelength, the numerical aperture (NA), and the coherence of the illumination source. To pattern features with smaller dimensions, NA is increased every technology node. The NA of lithography equipment used in the 65-nm node is higher than that of the 90-nm node (65-nm NA $=0.9-1.2$; 90-nm NA$=0.7-0.85$). The OPR, which determines the number of neighbor causing CD variation of border poly, decreases with higher NA.[8] For example, OPR decreases 21% as NA increases from 0.75 to 0.95, for a given set of illumination settings. Effectively, the scaling in OPR is somewhat slower than design rule scaling. (We assume that design rule scaling from 90 to 65 nm is in the range of 25 to 30%. In addition, design rule scaling to 45 nm is supposed to 50%.) AP-to-border poly spacing thus needs to be increased compared to that of the 90-nm node. On the other hand, OPR scaling from 90 to 45 nm is only 37%, which is much slower than design rule scaling. However, most standard-cell libraries at 45 nm have a dummy poly between border poly and cell outline for reducing interaction from neighboring cells. As AP is placed between standard cells that have the dummy poly, AP may shield the proximity effect without increase of AP-to-border poly spacing. We believe that the area penalty induced by AP is not significant, even with design rule scaling. Furthermore, chip size does not change using our intelligent placement optimization, which we describe next.

3.3 *Postplacement Perturbation for Improved Auxiliary Pattern Insertion*

The presence of an AP in close proximity to another AP corresponding to a different cell may violate minimum spacing rules for some placement configurations. This may inhibit insertion of AP for cells in such configurations. Hence, we propose to insert AP at the design level by perturbing the detailed placement. These perturbations do not increase chip size, since they simply take advantage of (by repartitioning) existing white space of the standard-cell placement. In this section, we describe a new detailed-placement perturbation algorithm using various types of AP. This approach extends the algorithm presented by Gupta, Kahng, and Park[9] to handle all three types of AP.

Define S_a^{AL} to be the space between the left outline of the cell and the active geometry, and S_{a-1}^{AB} to be the space between the right outline of the cell and active layer. Similarly, let S_a^{PL} be the space between the left outline of the cell and the poly, and S_{a-1}^{PR} be the space between the right outline of the cell and poly layer. S_a^L and S_{a-1}^R are defined as follows.

$$S_{a-1}^R = \min\{(S_{a-1}^{AR_1}, \ldots, S_{a-1}^{AR_n}), (S_{a-1}^{PR_1}, \ldots, S_{a-1}^{PR_n})\},$$

$$S_a^L = \min\{(S_a^{AL_1}, \ldots, S_a^{AL_n}), (S_a^{PL_1}, \ldots, S_a^{PL_n})\}. \quad (1)$$

Assume a set $AS=AS_1, \ldots, AS_m$ of spacings that are "AP-correct," i.e., if the spacing of boundary shapes between cells belongs to the set AS, then the required number of APs can be inserted between cells. For example, AS_1 and AS_2 are the required spacings for one AP and two APs, respectively. Figure 8 shows an example portion of the in-

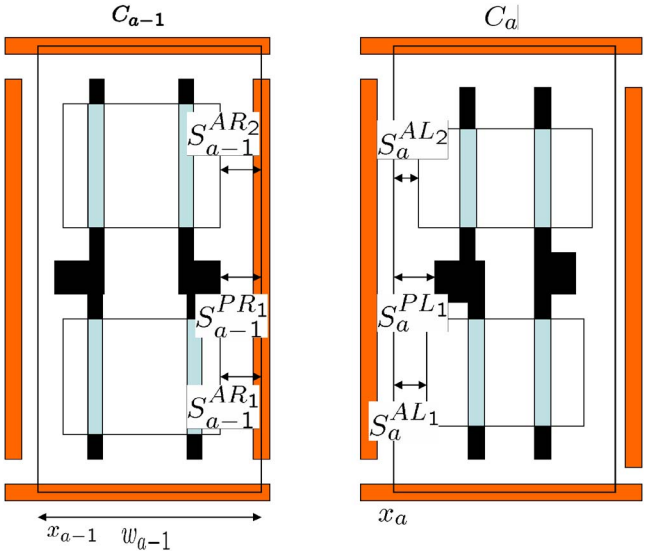

Fig. 8 An example for an algorithm of postplacement optimization.

put for our postplacement optimization algorithm. Let W_a denote the width of the cell C_a, and let x_a and x_a^i denote the (leftmost) placement coordinates of the original standard cell and the modified standard cell with type-i AP, respectively. Let δ denote a placement perturbation by which the modified standard cell will have an AP-correct spacing. Then the **AP-correct placement perturbation problem** may be formulated as:

$$\text{minimize } \sum |\delta_i|$$

$$\text{subject to } \delta_a + x_a^i - \delta_{a-1} - x_{a-1}^i - W_{a-1} + S_a^L + S_{a-1}^R \in AS.$$

Our objective is to minimize total placement perturbation from the original cell location and area penalty. We solve for the perturbed placement locations of the cells using a dynamic programming recurrence. We solve this "continuous" version of the prior problem with the following dynamic programming recurrence:

$$\text{cos}(1,b) = |x_1^i - b|$$

$$\text{cost}(a,b) = \lambda(a)|(x_a^i - b)| +$$

$$\min_{j=x_{a-1}^i-\text{SRCH}}^{x_{a-1}^i+\text{SRCH}} \{\text{cost}(a-1,j) + \text{APcost}(a,b,a-1,j)\}. \quad (2)$$

Cost$(1,b)$ is the cost of placing the first cell of each standard-cell row at placement site number b. Cost(a,b) is the cost of placing cell a at placement site number b. The cells and the placement sites are indexed from left to right in the standard-cell row. We restrict the perturbation of any cell to *SRCH* placement sites from its initial location for timing-driven placement. *APCost* is the measure of total expected CD degradation of the vertically oriented poly geometries closest to the cell boundary at the worst defocus value for the cell. *APCost* depends on the space between border polys. If the space is smaller than the required spacing for one AP, *APCost* is infinite, since it causes overlap between APs. The method of computing *APCost* is shown in Fig. 9.

The modified cell placement corresponding to a feasible set of AP insertions can then be incorporated into a modified standard-cell GDSII. Cell definition in design exchange format (DEF) is changed according to the standard-cell GDSII used during postplacement optimization. For example, NAND2X2_T1_T3 is a new cell definition in DEF with type-1 V-AP at left outline and type-3 V-AP at the right outline of NAND2X2. Thus, the proposed placement optimization can modify the standard-cell placement and is consistent with the set of available APs for each cell.

3.4 *Modified Design Flow*

Figure 10 shows the flow sequence for AP generation and placement perturbation of instances. A standard-cell layout is input to an AP generation step, and then to an SRAF insertion step. The resulting layout is input to an OPC insertion step, which results in a set of OPC'ed standard-cell layouts corresponding to the master cells. These OPC'ed cell layouts will be instantiated within the final layout, according to the results of postplacement optimization. The AP-correct placement takes the OPC'ed standard-cell layout as an input. A final cell-based OPC layout is generated from the modified AP-correct placement and the OPC'ed standard-cell layouts.

APCost(a,b,a-1,j) of Cell C_a

Input:
Origin x (left) coordinate and length of cell C_a = b
Origin x (left) coordinate and length of cell C_{a-1} = j
Width of cell $C_a = w_a$
Width of cell $C_{a-1} = w_{a-1}$

Output:
Value of $APCost$

Algorithm:
01.**Case** $a = 1$: $APCost(1,b) = 0$
02.**Case** $a > 1$ **Do**
/* For three AP types for left and right outline, calculate weight according to boundary geometries. */
03. $space = x_a^i - x_{a-1}^i - W_{a-1} + S_a^L + S_{a-1}^R$
04. **if**$(space \neq AS)$ $weight = \infty$
05. **else** $weight = space$
06. $APCost(a,b,a-1,j)$ += $weight$

Fig. 9 *APCost* calculation.

4 Cell Characterization Considering Lithography Effects

AP-based OPC achieves substantial reduction in edge placement error (EPE) over COPC at any given focus condition. To demonstrate the timing and leakage impact of AP-based OPC and COPC, we perform lithography-aware cell characterization. In the rest of this section, we discuss details of this flow.

A significant fraction of across-chip linewidth variation is caused by linewidth change depending on poly line pitch, poly line shape (corners, jogs, etc.), and their orientations. Printed poly shape varies as a function of focus, exposure

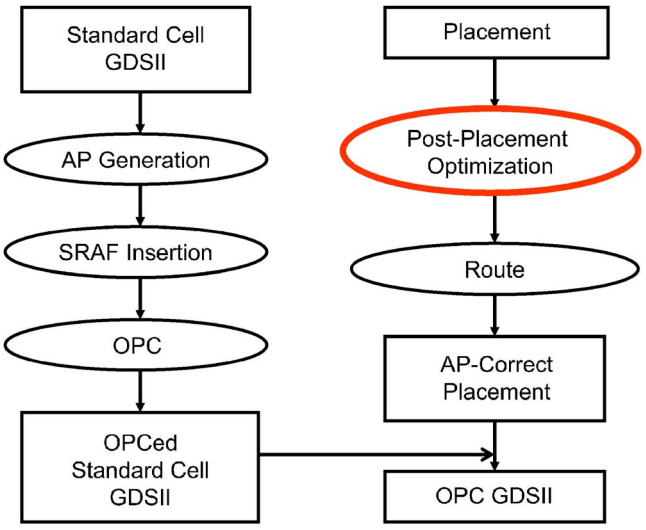

Fig. 10 Block diagram of a system for AP generation and placement perturbation of layout objects.

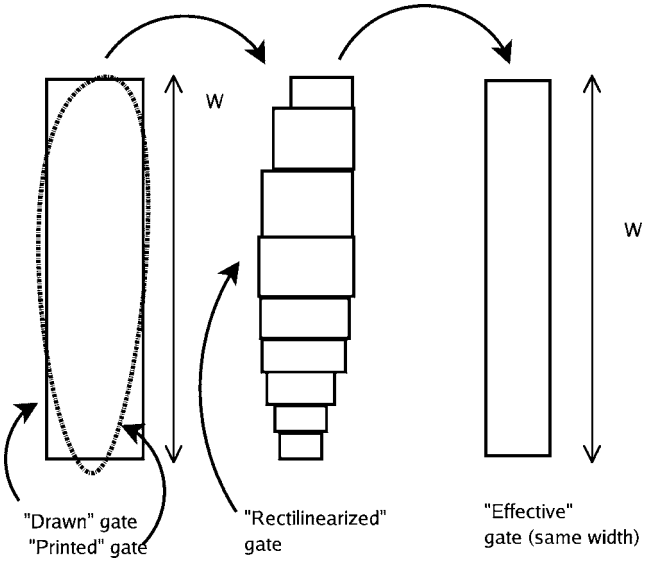

Fig. 11 Calculation of L_{avg} for timing and leakage from nonuniform geometry device.

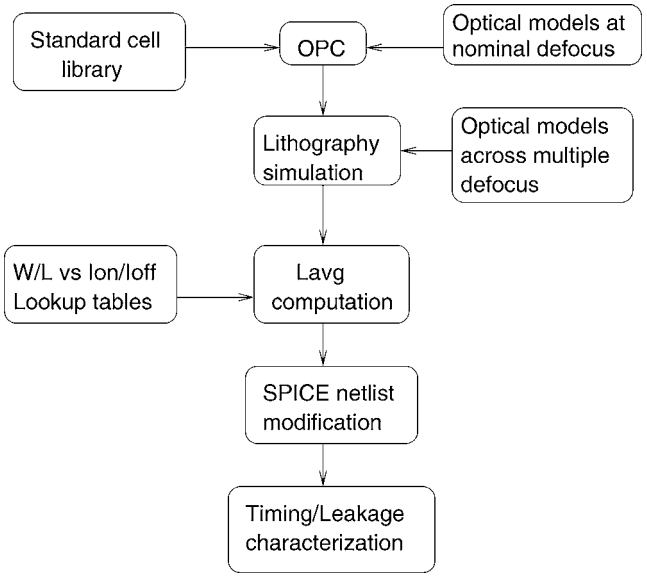

Fig. 12 Litho-aware standard-cell characterization flow.

dose, and layout parameters within the process window. In addition to linewidth (i.e., gate CD), field poly length, gate width, and contact enclosure may also change. However, these do not affect electrical parameters (i.e., delay and leakage) significantly. Delay is partially determined by saturation current and decreases linearly with decrease in linewidth. Subthreshold leakage increases exponentially with decrease in linewidth. Since linewidth is the smallest dimension related to devices, its variation translates to significant performance and leakage variability. Consequently, we focus only on characterization of gate CD impact in our litho-aware analysis.

4.1 *Average Gate Critical Dimension Computation*

SPICE simulations can be performed to characterize timing and leakage profiles of a standard cell using printed gate CD. However, existing device models for SPICE can only handle rectangular transistors, while printed devices have nonrectangular geometry.[13] The post-litho timing analysis flow presented in Ref. 6 considers CD at the center of the device and uses it as a representative value for the entire device. However, this is not accurate, since I_{on} and I_{off} of a device depend on its CD profile. To account for the gate CD profile using existing device models, we compute the *average* gate length for each device. I_{on} and I_{off} have different sensitivities to the same gate CD profile. Hence, we compute L_{avg} differently for timing and leakage. To compute L_{avg} of nonuniform geometry devices, we use the method outlined by Heng, Lee, and Gupta.[10] Their basic flow proposed in the paper takes in a gate shape contour (from lithography simulation) and performs rectilinearization. In this step, the nonuniform geometry is divided into multiple small rectangles with different W and L, as shown in Fig. 11. Separately, lookup tables for device I_{on} and I_{off} are created for different W and L combinations from SPICE simulations. I_{on} and I_{off} of the nonuniform geometry device are computed by summing up the corresponding values for each rectilinear (small) device from the lookup tables. L_{avg} of the actual printed gate contour is the gate length of a rectangle of the same gate width that yields the same on- or off-current (done by reverse lookup in the I_{on}/I_{off} table). This methodology yields $L_{avg,timing}$ and $L_{avg,leakage}$ corresponding to timing and leakage, respectively, and accounts for the nonuniformity in gate CD along the width of the gate.

4.2 *Lithography-Aware Cell Characterization*

The values of L_{avg} computed for each device in a standard cell are now used for accurate postlitho timing and leakage characterization. Standard cell SPICE netlists specify device names and their width and length (W/L) only. Positional information of devices is absent in the SPICE netlist. To associate printed CD of devices to their names, we run layout-versus-schematic (LVS) on standard-cell layouts to obtain their locations. Using LVS information, we update SPICE netlists with L_{avg} gate lengths computed from the rectilinearization of printed gate shapes. We create two versions of the SPICE netlist: one for timing characterization (updated with $L_{avg,timing}$) and the other for leakage characterization (updated with $L_{avg,leakage}$). The complete litho variation-aware cell characterization flow is summarized in Fig. 12.

5 Experiments and Results

In this section, we describe our experimental setup to 1. compare the printabilities (in terms of EPE count) of MBOPC, COPC, and AP-based OPC; 2. demonstrate improvement in timing and leakage variability of AP-based OPC over COPC; and 3. demonstrate comparable timing and leakage variabilities of AP-based OPC and MBOPC.

5.1 *Experimental Setup*

To compare MBOPC and AP-based OPC, we first prepare two designs (AES and ALU128) from `opencores.org` for application of OPC. The circuits are synthesized using *Synopsys Design Compiler v2003.06-SP1*[11] with tight tim-

Table 1 AP insertion error for five different row utilizations across different postplacement optimizations. "Typical" corresponds to the original placement. "T3" and "All" represent AP-correct placements with type-3 AP and all types of AP, respectively. For utilizations <70%, postplacement optimization improves AP applicability to 100%.

Utilization (%)	90			80			70			60			50		
Flow	Typical	T3	All	Typical	T3	All	Typical	T3	All	Typical	T3	All	Typical	T3	AH
AES	9115	2512	1925	3199	68	55	3166	0	0	1873	0	0	1589	0	0
ALU	5613	3099	2542	2085	219	179	1670	0	0	727	0	0	813	0	0

ing constraints and a set of 50 most frequently used cells in the Artisan TSMC 90-nm library. AES and ALU128 are synthesized to 11,553 and 8572 cells, respectively. The synthesized netlists are then placed with row utilization ranging from 50 to 90%. On the lithography side, *Mentor Graphics Calibre v9.3 5.11*[12] is used for model-based OPC, assist feature insertion, and optical rule checking (ORC). Vector aerial image simulation is performed with wavelength $\lambda=193$ nm and NA=0.7 for 90 nm. Annular illumination with $\sigma=0.85/0.57$ is used. Our OPC setup conforms to those used in industry-strength recipes. To evaluate EPE for each type of OPC, we first perform MBOPC on the entire design using the setup described before. For AP-based OPC, we implement the flow described in Sec. 3.4.

To compare the timing and leakage variabilities of different OPC types at the cell level, we compare isolated and layout contexts of standard cells. The isolated context refers to the stand-alone version of the cell, and the layout context refers to the standard cell in a placement context. The layout context is constructed by placing copies of a given standard cell on all its four sides, to simulate OPE inside the center cell. We then perform: 1. cell based OPC without AP [denoted as COPC(WO)]; 2. AP-based OPC with vertical-only AP [denoted as COPC(V)]; 3. AP-based OPC with horizontal and vertical AP [denoted as COPC(HV)]; and 4. model-based OPC (MBOPC) on both versions of all chosen standard cells. We then perform lithography simulation at nominal and 100 nm defocus. We then execute the litho-aware characterization flow described in Sec. 4.

At the design level, comparison of timing and leakage variabilities from different types of OPC is not straightforward. To evaluate the necessity for performing chip-level post-lithography timing and leakage power analysis, we first evaluate gate poly EPE. AP-based OPC can be used as replacement for MBOPC without incurring performance degradation (due to CD variation), while achieving significant savings in OPC runtime. The litho quality achieved by MBOPC is an upper bound on that achieved by AP-based OPC measured in terms of EPE. This OPC runtime versus CD tradeoff can be utilized in a design-aware fashion to minimize design performance and power impact while improving OPC runtime. For instance, MBOPC can be applied to all timing-critical features, and AP-based OPC can be applied to all nontiming-critical features. To explore this runtime versus performance impact tradeoff, we perform MBOPC and AP-based OPC on different fractions of cells at layout. The choice of cell instances for performing MBOPC is determined by their timing criticality. The total OPC runtime is the sum of MBOPC runtime on all timing-critical cell instances, and the runtime of MBOPC for individual masters that are instantiated in the design.

To run MBOPC on timing-critical cells in the design, we first perform timing analysis on the design to identify cell instances on paths with slacks within 10, 20, 30, and 40% of clock cycle time. We then create a cover layer on all timing-critical cells in the design layout and run MBOPC only on the identified cells. OPC on the entire layout is completed by substituting AP OPC'ed cells into the layout. The flow discussed before creates a "timing criticality-aware" OPC solution of the layout. We refer to this solution as *hybrid OPC*. To compare this with a pure AP-based OPC solution, we substitute AP OPC'ed masters for all cells in

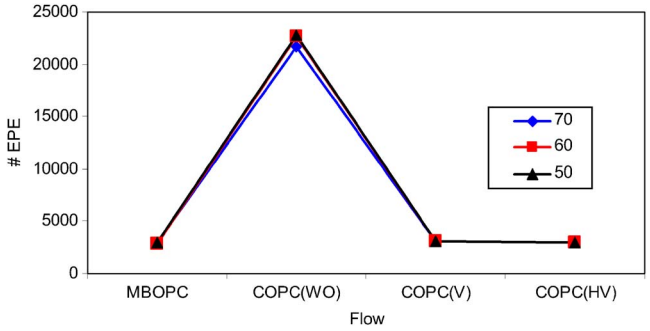

Fig. 13 EPE count of gate with various OPC methods for each of three different utilizations: COPC(WO) is a cell-based OPC without AP. COPC(V) is a cell-based OPC with only vertical AP. COPC(HV) is a cell-based OPC with H- and V-APs.

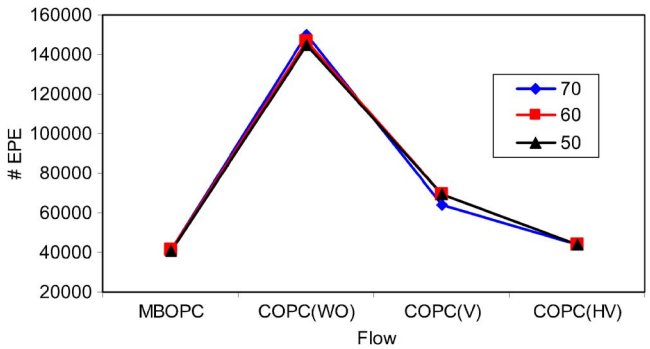

Fig. 14 EPE count of poly lines of AES design for three different row utilizations.

Table 2 Printability (in terms of EPE), OPC/ORC runtime and post-OPC GDSII file size for different types of OPC. COPC(HV) improves EPE over COPC(WO) by an average of 68%. Poly EPE count of COPC(HV) matches that of MBOPC within 6%. For AES, COPC(HV) reduces OPC runtime over MBOPC by a factor of 42×.

Design	Utilization (%)	Flow	EPE (Gate)	EPE (Poly)	GDSII size (MB)	OPC runtime (sec)	ORC runtime (sec)
AES	70	MBOPC	6972	41,365	3826	7932	943
		COPC(WO)	37,682	150,300	741	144	543
		COPC(V)	7528	63,942	776	168	598
		COPC(HV)	7240	44,110	789	192	621
	60	MBOPC	6988	41,043	3823	7943	940
		COPC(WO)	36,649	146,574	743	144	547
		COPC(V)	7522	69,198	780	168	599
		COPC(HV)	7290	44,023	799	192	641
	50	MBOPC	6974	40,636	3811	7943	938
		COPC(WO)	36,496	144,382	740	144	547
		COPC(V)	7509	69,198	786	168	602
		COPC(HV)	7217	44,012	799	192	641
ALU	70	MBOPC	2895	30,029	3213	4109	772
		COPC(WO)	21,675	86,926	721	120	364
		COPC(V)	3076	39,988	745	136	394
		COPC(HV)	2947	31,323	774	160	410
	60	MBOPC	2827	29,751	3221	4109	777
		COPC(WO)	22,711	92,740	722	120	373
		COPC(V)	3092	39,481	744	136	389
		COPC(HV)	2964	31,101	776	160	410
	50	MBOPC	2949	29,446	3222	4121	778
		COPC(WO)	22,823	91,946	703	120	376
		COPC(V)	3036	45,012	742	136	399
		COPC(HV)	2981	31,323	776	160	411

the design to create an AP OPC'ed GDS. We then perform ORC to evaluate gate EPE.

5.2 *Experimental Results*

We evaluate the quality of AP-based OPC by comparing it with MBOPC. The criteria chosen for evaluation are 1. AP insertion error, 2. OPC metrics (EPE, OPC runtime, file size), and 3. leakage and timing spread. AP insertion error is defined as the number of vertical edges of standard cells in which AP cannot be inserted, even after postplacement optimization. Table 1 shows the AP insertion error for five different utilizations and for three different placement contexts: 1. typical cell placement, 2. optimized cell placement with only type-3 AP, and 3. optimized cell placement with all combinations of AP. For row utilizations that are $<70\%$, postplacement optimizations can achieve 100% AP applicability without increasing chip size. Postplacement optimization with all combinations of AP can reduce AP insertion error over optimization with type-3 AP by an average of 20% for utilizations greater than 70%.

To evaluate the impact of AP-based OPC on printability, we perform ORC on gate and field poly and measure EPE count. For this study, we perform ORC to flag all layout edge fragments with error greater than 10% of drawn CD at the worst defocus condition. Figure 13 shows the EPE count of gates of the ALU design with various OPC meth-

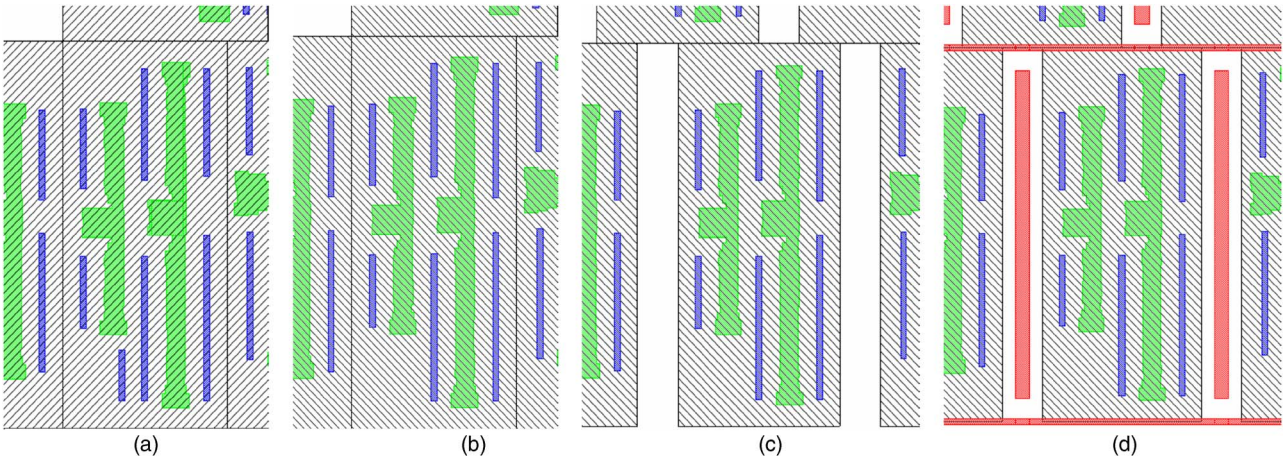

Fig. 15 Layouts with various OPC methods: (a) MBOPC, (b) COPC with no placement optimization, (c) COPC with placement optimization, and (d) COPC with placement optimization and AP. Red, blue, and green colors represent AP, SBAR, and OPC geometries, respectively. (Color online only.)

ods. EPE count of two AP-based OPC methods match that of MBOPC within 3%. Figure 14 shows the EPE count of poly lines of an AES test case. EPE count of OPC with only V-AP is 35% more than that of MBOPC. This is because of poly line-end shortening due to OPE between cell rows. However, EPE count of AP-based OPC with H- and V-APs match that of MBOPC within 6%. This also corresponds to an average improvement of 68% over COPC without AP [COPC(WC)]. We compare the average CD difference of devices near cell outline for three cases of COPC with MBOPC. The average CD differences for 1. COPC with no placement optimization, 2. COPC with placement optimization, and 3. COPC with placement optimization and AP over MBOPC are 7.2, 2.5, and 1.2 nm, respectively. Figure 15 shows the actual layouts with various OPC methods.

OPC runtimes for MBOPC, COPC(WO), COPC(V), and COPC(HV) are summarized in Table 2. OPC runtime denotes the runtime of assist feature insertion, MBOPC, and AP insertion (in the case of AP-based OPC). The AES and ALU designs use 48 and 40 standard-cell definitions, respectively. From the table, we can observe that COPC (WO, V, and HV) improves the runtime by an order of magnitude versus MBOPC. The improvement will be more apparent as the design size increases. From the table, we can observe that COPC (WO, V, and HV) runtimes are comparable between AES and ALU test cases. But we can clearly see the sharp rise in MBOPC runtime as the number of instances increases from 8572 (ALU) to 11,553 (AES). COPC(HV) reduces runtime over MBOPC by 42× and by 25× for AES and ALU, respectively. We can also observe reduction of GDSII file size and ORC runtimes. COPC maintains the original cell hierarchy, thereby reducing GDSII file size and ORC runtime over MBOPC.

Table 3 shows the percentage spread in leakage and timing of eight standard cells at nominal defocus. Leakage spread at any given focus condition is the percentage change in cell leakage power between the isolated and the layout context of the cell. Timing spread is computed as the percentage change in the rise delay of the cell output pin at a fixed load capacitance and slew condition. (In our experiments, we measured delay values at a load capacitance of 6.5 pF and a transition time of 140 ps.) From the table, we can observe that COPC(HV) improves leakage variability over COPC(WO) by an average of 65%. COPC(HV) improves timing variability over COPC(WO) by an average of 42%. Another important trend apparent from the results is that the leakage and timing spread of COPC(HV) and MBOPC are comparable to within one percentage point for all of the cells.

Table 4 shows the comparison between gate EPE count between hybrid OPC solution and a pure AP-based OPC solution for the AES test case with 70% row utilization. The number of timing-critical cells in the design, based on different timing slack criteria, is also shown. Gate EPE count is the number of edge fragments on border poly geometries that have greater than 3-nm EPE at the best focus level. From the table, we can observe that hybrid OPC runtime increases in proportion to the number of cells for which MBOPC is applied. For AP-based OPC, the total

Table 3 Comparison of leakage and timing spread of standard cells between WO [i.e., COPC(WO)], HV [i.e., COPC(HV)], and MB(i.e., MBOPC), COPC(HV) improves leakage variability over COPC(WO) in the range 1 to 92% and timing variability in the range 1 to 85%

Cell	Percent leakage variation			Percent timing variation		
	WO	HV	MB	WO	HV	MB
and3x1	8.17	1.99	2.96	4.09	0.59	0.67
invx2	9.67	2.55	2.32	2.48	0.42	1.34
mx2x1	0.35	1.63	2.86	4.31	1.24	2.85
nand2bx1	7.06	2.21	3.38	0.51	1.51	1.13
nand2x2	8.48	0.64	1.69	0.82	0.71	1.23
nor2x2	10.65	1.58	2.05	0.66	0.66	0.44
nor4x2	10.20	1.03	2.14	1.22	0.37	0.27
xor2x1	1.77	1.66	1.55	0.65	1.07	0.72

Table 4 Gate EPE count for hybrid OPC for different fraction of timing-critical cells in AES test cases implemented in TSMC 90-nm technology. Hybrid OPC runtime is proportional to the number of timing-critical cells. Gate EPE count for hybrid OPC as well as COPC(HV) are within 0.1% of each other.

Timing slack (as a percent of cycle time)	Number of cells within timing slack	Hybrid OPC runtime (s)	Gate EPE count (hybrid OPC)
10	5640	36,930	1471
20	6369	42,588	1472
30	9165	60,837	1480
40	9221	61,329	1480

gate EPE count and runtime are independent of the number of timing-critical cells and are 1479 and 326 sec, respectively. From the gate EPE count trend, we can observe that MBOPC and COPC(HV) achieve similar EPE on the gate poly (and) consequently, similar CD control). This eliminates the need for design level postlitho timing and leakage power analysis.

6 Conclusions

We propose a novel auxiliary pattern (AP)-based cell OPC method that has the OPC TAT advantages of COPC and printability performance comparable to that of MBOPC. Using a timing-aware DP-based method that perturbs detailed cell placements, we demonstrate a method for opportunistic insertion of AP at the full-chip level to maximize the benefits of AP-based OPC. Our AP-based OPC approach has shown a factor of 42× reduction in OPC runtime compared to MBOPC. The runtime advantage will be substantially higher for larger designs. Printability analysis of AP-based OPC shows that V-AP and V/H-AP can match gate EPE count of MBOPC within 3%. This is an improvement of 68%, on average, over cell-based OPC without APs. Our postplacement optimization method can achieve 100% AP applicability in designs with utilization less than 70%. For designs with utilization greater than 70%, we can achieve up to 80% AP applicability. Our proposed DP-based perturbation approach is timing aware; it does not modify the placement (and consequently routing) of timing-critical cells in the design, thereby preserving timing. Using a litho-aware timing and leakage analysis flow, we demonstrate 65 and 42% reductions in timing and leakage variabilities, respectively, over cell-based OPC. Further, the spread in leakage and timing match those of MBOPC within 1%. This demonstrates that adoption of AP-based OPC does not degrade design performance and power. AP-based OPC can be adopted in an industrial flow with significant runtime savings without any performance degradation.

References

1. Y. Granik, N. B. Cobb, and T. Do, "Universal process modeling with VTRE for OPC," *Proc. SPIE* **4691**, 377–394 (2002).
2. P. Gupta, F. L. Heng, and M. Lavin, "Merits of cellwise model-based OPC," *Proc. SPIE* **5379**, 182–189 (2004).
3. X. Wang, M. Pilloff, H. Tang, and C. Wu, "Exploiting hierarchical structure to enhance cell-based RET with localized OPC reconfiguration," *Proc. SPIE* **5756**, 361–367 (2005).
4. C. H. Wallace and C. H. Jang, "Sub-resolution assist features for photolithography with trim ends," U.S. Patent Application No. 20070128525.
5. M. Garg, A. Kumar, I. van Wingerden, and L. Le Cam, "Litho-driven layouts for reducing performance variability," *Proc. IEEE Intl. Symp. Circuits Syst.*, pp. 3551–3554 (2005).
6. J. Yang, L. Capodieci, and D. Sylvester, "Advanced timing analysis based on post-OPC extraction of critical dimensions," *Proc. IEEE/ACM Intl. Design Auto. Conf.*, pp. 359–364 (2005).
7. K. Cao, S. Dobre, and J. Hu, "Standard cell characterization considering lithography induced variations," *Proc. ACM/IEEE Design Auto. Conf.*, pp. 801–804 (2006).
8. A. K. Wong, *Resolution Enhancement Techniques in Optical Lithography*, Vol. **TT47**, SPIE Press, Bellingham, WA (2001).
9. P. Gupta, A. B. Kahng, and C. H. Park, "Detailed placement for improved depth of focus and CD control," *Proc. Asia South Pacific Design Auto. Conf.*, pp. 343–348 (2005).
10. F. L. Heng, J. F. Lee, and P. Gupta, "Toward through-process layout quality metrics," *Proc. SPIE* **5756**, 161–167 (2005).
11. Synopsys Design Compiler User's Manual, see http://www.synopsys.com/.
12. Mentor Graphics Calibre RET User's Manual, see http://www.mentor.com.
13. A. B. Kahng and C. H. Park, "Auxiliary pattern for cell-based OPC," *Proc. SPIE* **6349**, 63494S (2006).

Andrew B. Kahng received his AB degree in applied mathematics (physics) from Harvard College, and his MS and PhD degrees in computer science from the University of California at San Diego, La Jolla. He joined the UCLA Computer Science Department as an assistant professor in July 1989, and became associate professor in July 1994 and full professor in July 1998. In January 2001, he joined UCSD as Professor in the CSE and ECE Departments. He served as Associate Chair of the UCSD CSE Department from 2003–2004. Professor Kahng was the founding General Chair of the 1997 ACM/IEEE International Symposium on Physical Design, co-founder of the ACM Workshop on System-Level Interconnect Prediction, and defined the physical design roadmap as a member of the Design Tools and Test Technology Working Group (TWG) for the 1997, 1998 and 1999 renewals of the International Technology Roadmap for Semiconductors. From 2000 through 2003, he was Chair of both the U.S. Design Technology Working Group, and of the Design International Technology Working Group, and continues to serve as co-chair of the Design ITWG. He has been an executive committee member of the MARCO Gigascale Systems Research Center since its inception in 1998. In October 2004, Professor Kahng co-founded Blaze DFM, Inc. and served as CTO of the company until resuming his duties at UCSD in September 2006. Professor Kahng has published over 300 journal and conference papers. He has received NSF Research Initiation and Young Investigator awards, 11 Best Paper nominations, and 6 Best Paper awards. Since 1997, his research in IC design for manufacturability has pioneered methods for automated phase-shift mask layout, variability-aware analyses and optimizations, CMP fill synthesis, and parametric yield-drawn, cost-driven methodologies for chip implementation.

Swamy Muddu received his BEng degree in electronics and communication engineering from the College of Engineering, Osmania University, Hyderabad, in 2001, and MS degree from the ECE department at UC San Diego in 2003. He is currently a candidate for the PhD. During 2005, he was an engineer at Blaze DFM, Sunnyvale, California. His research interests are in the area of VLSI physical design for manufacturability, modeling of manufacturing variation, and low-power design.

Chul-Hong Park is a PhD student in the ECE department at UC San Diego. He received the BS and MS degrees in mathematics from Kyung Hee University, Seoul, in 1992 and 1994, respectively. He worked at CAE, Semiconductor R&D Center, Samsung Electronics, from 1994 to 2003. He received 1998 Honorable Outstanding Researcher Award and two Best Paper awards at Samsung Electronics. He has published more than 20 papers and has seven patents. He is currently a candidate for the PhD. His research interests include lithographic design for yield and VLSI design-manufacturing interface.