Auxiliary Pattern for Cell-Based OPC

Andrew B. Kahng[a,b,c] and Chul-Hong Park[a]

[a]UCSD ECE Department, La Jolla, CA
[b]UCSD CSE Department, La Jolla, CA
[c]Blaze DFM, Inc., Sunnyvale, California

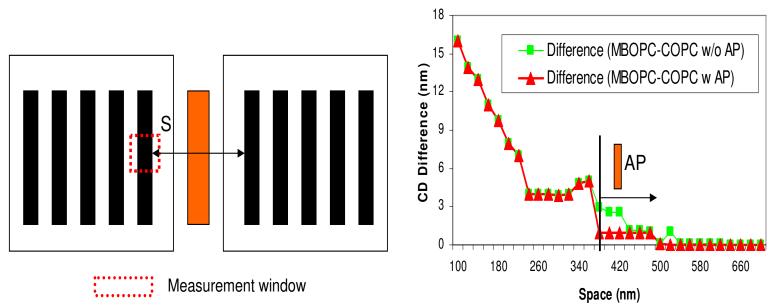
ABSTRACT

The runtime of model-based optical proximity correction (OPC) tools has grown unacceptably with each successive technology generation, and has emerged as one of the major bottlenecks for turnaround time (TAT) of IC data preparation and manufacturing. 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 inter-cell optical interactions that affect feature printability in a layout context. In this work, we propose *auxiliary pattern-enabled* cell-based OPC which can minimize the CD differences between cell-based OPC and model-based OPC. To enable effective insertion of auxiliary pattern (AP) in the design, we also propose a post-placement optimization of a standard cell block with respect to detailed placement. By dynamic programming-based placement perturbation, we achieve 100% AP applicability in designs with placement utilizations of $<$ 70%. In an evaluation with a complete industrial flow, cell-based OPC with AP can match gate edge placement error (EPE) count of model-based OPC within 4%. This is an improvement of 90%, on average, over cell-based OPC without APs. The AP-based OPC approach can reduce OPC runtimes versus model-based OPC by up to 40X in our benchmark designs. We can also achieve reduction of GDSII file size and ORC runtimes due to hierarchy maintenance of cell-based OPC.

1. INTRODUCTION

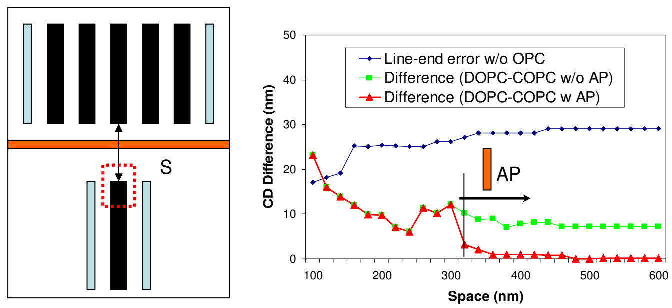
Optical proximity correction (OPC) has been a key enabler of the aggressive integrated-circuit (IC) technology scaling implicit in Moore's Law. OPC determines the photomask patterns that enable drawn layout features to be faithfully and accurately reproduced by optical lithography onto the wafer. The cell-based OPC (COPC) approach runs OPC once per each cell definition rather than once per placement or unique instantiation of each cell. In other words, in the cell-based OPC approach, the master cell layouts in the standard cell library are corrected before placement, and then placement and routing steps of IC design are completed with the corrected master cells. Unfortunately, optical proximity effects (OPEs) in lithography imply certain interactions 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 cell-based OPC solution can be incorrect when instantiated in a full-chip layout. As a result, there can be a large difference in feature critical dimension (CD) between COPC and model-based OPC (MBOPC).

Gupta *et al.*[2] proposed the use of dummy features to consider the different neighboring environments of a cell. The dummy features are inserted in predetermined areas before OPC is performed and then taken out from the layout after OPC to maintain the design rules, and poly-to-poly and poly-to-contact spacings, of the cell layout. However, the dummy features have the limited ability to represent the proximity effect from pattern geometries of neighboring cells and hence CD errors still remain after cell-based OPC. Wang *et al.*[5] describe a method which accounts for OPC re-correction in proximity interaction areas between cells of a standard-cell block. The approach is to first perform cell-based OPC, and then to stitch already-corrected cells into the layout, re-correcting the OPC solution within interacting areas at the cell boundaries, to obtain a proximity-corrected final layout. However, the stitching area can increase OPC runtime and degrade the ostensible benefits of cell-based OPC such as cell reuse and cell-based timing analysis. Matsunawa *et al.*[3] also introduce a genetic algorithm to correct the stitching area. However, the method also requires rework of the layout (e.g., GDSII representation) of the OPC'ed cell design.



(a) (b)

Figure 1. CD impact of AP on line width: maximum CD differences between COPC and MBOPC are 3nm without AP and 1nm with AP.



(a) (b)

Figure 2. CD impact of AP at line-end: maximum CD differences between COPC and MBOPC are 10nm without AP and 3nm with AP.

In this work, we devise a novel AP technique which *shields* poly patterns near the cell outline from proximity effects of neighboring cells. This enables the COPC approach to achieve the same printability as MBOPC methods. AP features consist of vertical (V-AP) and/or horizontal (H-AP) dummy poly lines. V-AP features print on wafer and are located within the same cell row; H-VP features do not print on wafer and are located in the overlap region between cell rows. We also propose a post-placement optimization methodology to improve AP applicability. Using this methodology, we improve the quality of the COPC solution to be even closer to that of the MBOPC solution. Our main contributions are as follows.

- We propose a novel approach for application of COPC to designs, based on insertion of auxiliary patterns. APs minimize CD difference between COPC and MBOPC. We demonstrate that COPC with V-AP and a combination of V- and H-AP can reduce edge placement error (EPE) by 90% versus typical COPC.
- AP insertion might not be feasible on all instantiations of a standard cell in the design. We propose a methodology for perturbation of detailed placement to allow maximal insertion of AP. We test our method within a complete industrial flow and achieve 100% AP applicability for COPC with design utilizations of $< 70\%$.

This paper is organized as follows. In Section 2 we discuss CD impact of AP in terms of line width, line-end and contact poly. In Section 3 we propose various types of AP. We discuss AP generation, printability impact of AP-based OPC and a placement perturbation method for improving feasibility of AP insertion. We discuss our experimental setup and results in Section 4. In Section 5, we summarize our contributions.

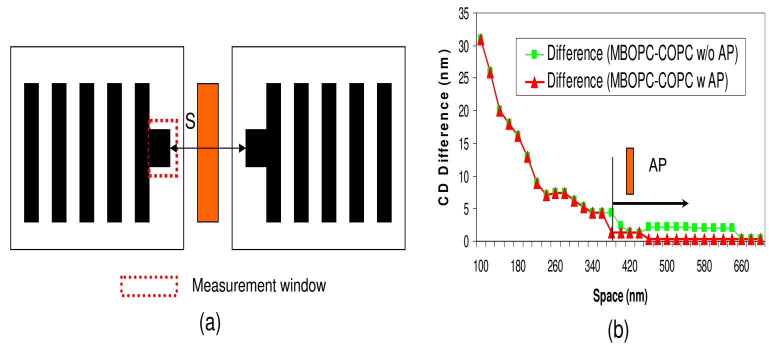


Figure 3. CD impact of AP in contact poly: maximum CD differences between COPC and MBOPC are 5nm without AP and 1nm with AP.

2. CD 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 CD impact of AP in terms of line width, line-end and contact poly*. Each test structure has two test cells which consist of line width of 0.1μm, pitch of 0.3μm and line length of 2.0μm.

Figure 1 shows the CD impact of AP on line width. In this plot, the x-axis gives the space between border polys of cells, and the y-axis indicates CD difference between MBOPC and COPC. We use AP width of 0.1μm, AP-to-poly space of 0.13μm and AP-to-AP space of 0.14μm. The AP can be inserted as long as the space between border polys is greater than 0.38μm. The maximum CD difference between MBOPC and COPC without AP is 3nm, while the maximum difference with AP is only 1nm. Proximity shield effect of AP with respect to line-end shortening is shown in Figure 2. We used horizontal AP width of 0.04μm for proximity shielding. The minimum space between line-end polys for insertion of APs is 0.32μm. The maximum CD difference between MBOPC and COPC without AP is 10nm, while the maximum difference with AP is about 3nm. 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 closet geometry to neighboring cells. The minimum space between contact polys for insertion of AP is 0.32μm, as shown in Figure 3. The maximum CD difference between MBOPC and COPC without AP is 5nm, while the maximum difference with AP is about 1nm. Consequently, COPC with AP achieves the same printability as MBOPC with respect to line patterning issues.

3. AP METHODOLOGY

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

3.1. AP Generation

Auxiliary patterns overcome the deficiencies of the COPC approach for standard cell layouts. The AP methodology processes an existing standard cell layout (e.g., input in GDSII) and then generates a new standard cell GDSII which contains AP as well as sub-resolution assist features (SRAFs). AP features consist of vertical (V-AP) and/or horizontal (H-AP) dummy poly shapes, as shown in Figure 4. 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.

Gates are typically laid out vertically (assuming horizontal cell rows). Since impact of optical proximity 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.

*Contact poly defines the overlapped area of poly and contact which may also be called "contact coverage".

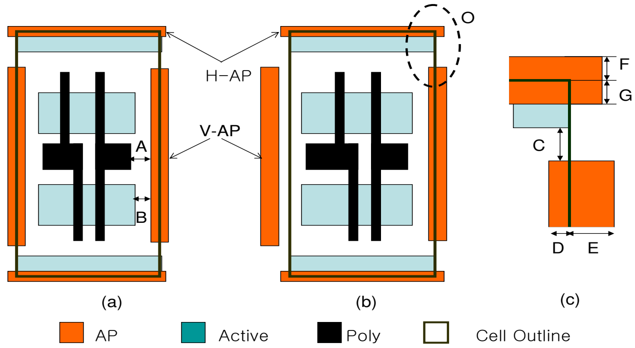


Figure 4. Examples of standard cell layout with APs: (a) Type-1 V-AP, (b) Type-2 V-AP and (c) an enlargement of the region O of (b).

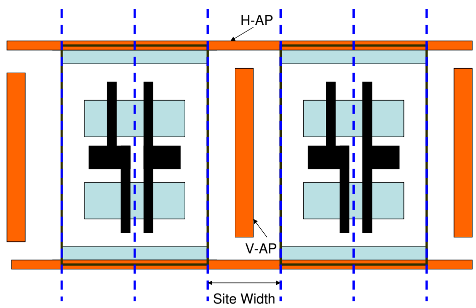


Figure 5: An example of standard cell layout with Type-3 V-AP.

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 horizontal-AP is as small as that of a SRAF. The H-AP differs from the SRAF technique because the location of the SRAF depends on the distance between poly lines, while the AP is exactly located at the cell boundary. In general, there is an active layer at the boundary between different cell rows, and hence the H-AP should not print on the wafer. There are three types of V-APs according to the location of insertion. We now describe the three types of vertical-APs as follows.

Type-1 V-AP Figure 4(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 Figure 4(c)) - being the distance from cell outline to left edge of AP - is the same as the right width of cell outline to right edge of AP (E in Figure 4(c)). Spaces A and B respectively define space between border poly and AP, and space between active-layer geometry and AP. The Type-1 AP achieves minimum area penalty during cell placement compared to other types of AP. Since A and B in the typical standard cell are smaller than required minimum spaces, 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. Thus, the restricted design rule approach[1] may also be used to modify cells with Type-1 V-AP for COPC.

Type-2 V-AP Figure 4(b) shows that in the case of violating the minimum poly-to-poly and poly-to-active spacing rules with the Type-1 V-AP, a Type-2 V-AP represents a practical means of integrating the AP construct within standard cell placements. Type-2 V-AP locations satisfy both A and B parameters of the minimum design rules. Width D is different from width E. The Type-2 V-AP can also be placed at the outside of the cell outline. In Figure 4(c), which is an enlargement of region O of Figure 4(b), C is the space between V-AP and active layer, and is the same as the design rule for minimum space between poly line-end and active layer. The width from cell outline to the bottom edge G of the H-AP is the same as the width from cell outline

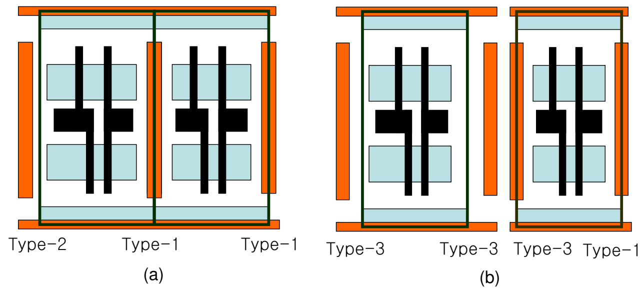


Figure 6. Standard cell layouts constructed by combinations of the three types of APs: (a) a two-cell layout with Type-1 and Type-2 V-APs, and (b) a two-cell layout with a combination of Type-1 and Type-3 V-APs.

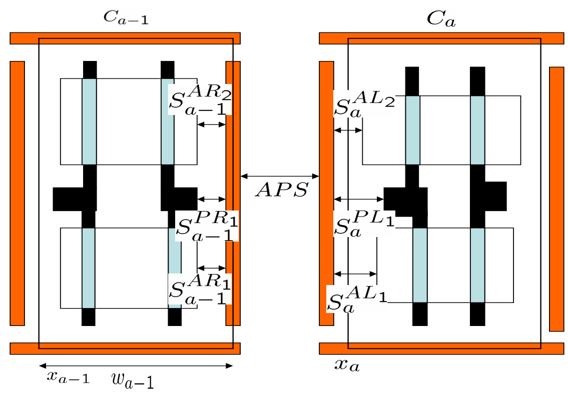


Figure 7: An example for algorithm of post-placement optimization.

to top edge F of the H-AP.

Type-3 V-AP Figure 5 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 AP can maintain minimum spacing rules for poly-to-poly and poly-to-active while simultaneously minimizing area penalty. For compatibility with design rule checks and other objectives, it can be beneficial to match a right end of a horizontal AP to a right edge of a right vertical AP, and/or to match a left end of a horizontal AP to a left edge of a left vertical AP.

Various auxiliary patterns can be constructed by combinations of the above three types of APs. Figure 6 shows two examples: (a) a two-cell placement with a combination of cells with Type-1 and Type-2 V-APs; and (b) 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, Figure 6(a) and Figure 6(b) show APs completely overlapped or having certain required spacing to each other, respectively.

3.2. Detailed Placement Perturbation for Improved AP Applicability

In this section, a proposed post-placement perturbation algorithm for auxiliary pattern correction. Since APs should have prescribed spacings to each other or be completely overlapped, a post-placement optimization step improves AP applicability[†].

Given a cell C_a, let AP_a^L and AP_{a-1}^R be the sets of valid AP geometries in the cell which are located at the left and right outlines of the cell respectively. The following discussion is with respect to the illustration given

[†]AP applicability requires that APs between adjacent two cells are completely overlapped or have certain required spacing, i.e., "AP-correct spacing", to each other.

in Figure 7. Define S_a^L to be the minimum space between the left outline of the cell and the AP corresponding to the cell. The space, AP_a^L, is determined by a minimum value of AP-to-poly space, S^{LP}, and AP-to-active space, S^{LA}. Similarly, define S_{a-1}^R to be the space between the right outline of the cell and AP_{a-1}^R. Also let the set APS consist of spaces (e.g., the two spaces [0, minimum poly-to-poly space]) which are "AP-correct".

Let W_a denote the width of 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:

$$\begin{aligned} &\text{Minimize} \sum \mid \delta_i \mid \\ &(\delta_a + x_a^i + S_a^R) - (\delta_{a-1} + x_{a-1}^i + S_{a-1}^L) \in APS \\ \text{s.t. } &S_a^R = \min \{(S_a^{AR_1}, ..., S_a^{AR_n}), (S_a^{PR_1}, ..., S_a^{PR_n})\} \\ &S_{a-1}^L = \min \{(S_{a-1}^{AL_1}, ..., S_{a-1}^{AL_n}), (S_{a-1}^{PL_1}, ..., S_a^{PL_n})\} \end{aligned}$$

Our objective is to minimize total placement perturbation from original cell locations as well as any area penalty. We solve for the perturbed placement locations of cells using a dynamic programming recurrence and cost function.

$$\begin{aligned} Cost(1,b) &= \mid x_1^i - b \mid \\ Cost(a,b) &= \lambda(a) \mid (x_a^i - b) \mid + \\ Min_{j=x_{a-1}^i - SRCH}^{x_{a-1}^i + SRCH} &\{Cost(a-1,j) + APCost(a,b,a-1,j)\} \end{aligned} \tag{1}$$

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:
Case $a = 1$: $APCost(1,b) = 0$
Case $a > 1$ **Do**
/* Three V-AP types are available for each left and right outline.
For $(k = 1 ; k = 3 ; k = k + 1)$\{ /* Three V-AP types for left outline
  For $(g = 1 ; g = 3 ; g = g + 1)$ \{ /* Three V-AP types for right outline
    $space(k,g) = S_{a-1}^R(k) - S_a^L(g)$ /* Space between APs
    if$(space(k,g) < APS)$ $weight(k,g) = \infty$
      else $weight(k,g) = space(k,g)$
    $APCost(a,b,a-1,j)$ += $weight(k,g)$
  \}
\}

Figure 8: *APCost* calculation.

$Cost(a,b)$ 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. This helps contain the delay and runtime overheads of AP-correct placement post-processing. $APCost$ corresponds to the printability of the vertically oriented poly geometries closest to the cell boundary, and depends on the space between APs. If the space is smaller than APS, $APCost$ is infinite since it affects pattern bridging between AP geometries. The method of computing APCost is shown in Figure 8.

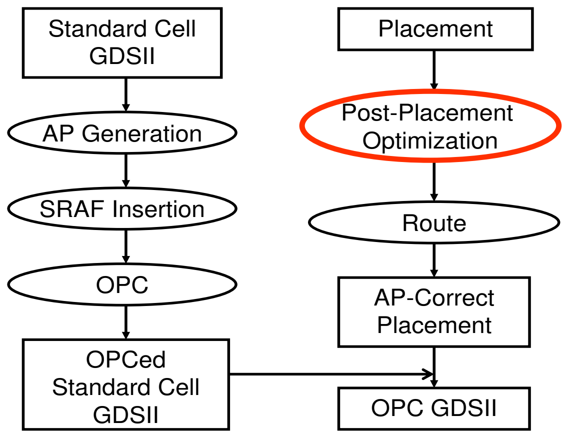


Figure 9: Block diagram of a system for AP generation and placement perturbation of layout objects.

The modified cell placement corresponding to a feasible set of AP insertions can be incorporated into a modified standard cell GDSII. Cell definition in DEF (Design Exchange Format) is changed according to the usage, i.e., instantiation, of master cell GDSII. 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 right outline of NAND2X2. Thus, the proposed placement optimization can modify the standard cell placement and consistent with the set of available APs for each cell, to improve the use of APs and printability metrics.

3.3. Modified Design Flow

To account for new geometric constraints that arise due to AP insertion in physical design, we add AP generation in standard cell GDSII as well as post-placement optimization into the current ASIC design methodology. Figure 9 shows the block diagram of a system for AP generation and placement perturbation of layout objects in a standard cell layout design. 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 post-placement optimization, i.e., to achieve a high-quality COPC solution. An original cell placement solution is perturbed by post-placement optimization to admit the insertion of any combination of AP types and the design rule-correct overlap or spacing of APs. The modified, 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.

4. EXPERIMENTS AND RESULTS

In this section, we describe our experimental setup to compare printability, runtime and GDSII size between model-based OPC and AP-based OPC at the design level.

4.1. Experimental Setup

To compare MBOPC and AP-based OPC at the design level, we first prepare two designs (AES and ALU128) from `opencores.org` for application of OPC. The circuits are synthesized using *Synopsys Design Compiler* with tight timing constraints and a set of 50 most frequently used cell models in the Artisan TSMC 90nm library. AES and ALU128 are synthesized to 11553 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* is used for model-based OPC, SRAF OPC and optical rule checking (ORC). Vector aerial image simulation is performed with wavelength λ = 193nm and NA = 0.85 for 90nm. An annular aperture with $\sigma = 0.96/0.76$ is used. The thicknesses of photoresist and bottom-ARC (anti-reflective coating) are 0.16μm and 0.036μm, respectively. Our

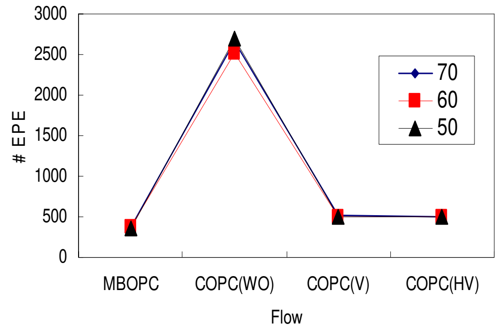


Figure 10. EPE count of gate with various OPC methods for each of three different utilizations: COPC(WO) is cell-based OPC without AP. COPC(V) is cell-based OPC with only vertical AP. COPC(HV) is cell-based OPC with H- and V-APs.

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 above. For AP-based OPC, we implement the flow described in Section 3.3.

4.2. 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 and (2) OPC metrics (EPE, OPC runtime, filesize). Table 1 shows AP insertion error for each five different utilizations and for three different placement contexts: (1) typical cell placement, (2) optimized cell placement with only Type-3 V-AP, and (3) cell placement with all combinations of AP. For row utilizations of $<$ 70%, post-placement optimizations can achieve 100% AP applicability. Post-placement optimization with all combinations of AP can reduce AP insertion error over optimization with Type-3 V-AP by an average of 20% for utilizations of $>$ 70%.

We have also validated the printability of AP-based OPC for gate and field poly. EPE count is the number of edge fragments having greater than 10% EPE at worst-case defocus level. Figure 11 shows the EPE count of gate of ALU design with various OPC methods. The EPE count of AP-based OPC with all types of APs matches that of model-based OPC to within an average difference of 4%. Figure 10 shows the EPE count of poly lines of the AES design. EPE count of OPC with only V-AP is 30% 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 matches that of MBOPC within 6%. This also corresponds to an average improvement of 90% over COPC without AP (WO).

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 case of AP-based OPC). AES and ALU designs used 48 and 40 of standard cell definitions, respectively. From the table, we can observe that COPC (WO and V/HV) improve the runtime by an order of magnitude versus MBOPC. This 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 testcases. But we can clearly see the sharp rise in MBOPC runtime as the number of instances increases from 8572 (ALU) to 11553 (AES). COPC(HV) reduces runtime over MBOPC by 40X and 25X for AES and ALU respectively. We can also observe reduction of GDSII size and ORC runtimes. COPC maintains the original cell hierarchy, thereby reducing GDSII size and ORC runtime over MBOPC.

5. CONCLUSIONS

We have proposed a novel auxiliary pattern-based cell OPC method that has performance comparable to model-based OPC with significant runtime advantage. The AP-based OPC approach achieves a factor of 40X reduction in OPC runtime compared to MBOPC. The runtime advantage will be substantially higher for larger designs.

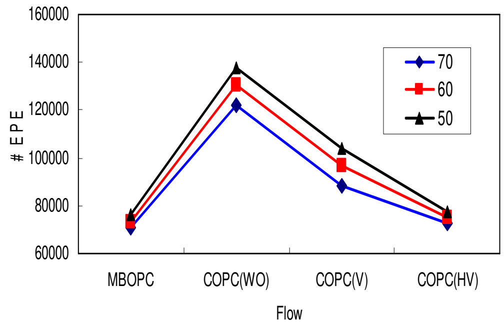


Figure 11: EPE count of poly lines of AES design for three different row utilizations.

Utilization (%)	90			80			70			60			50		
Flow	Typical	T3	All	Typical	T3	All	Typical	T3	All	Typical	T3	All	Typical	T3	All
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

Table 1. AP insertion error for five different row utilizations across different post-placement 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 of $< 70\%$, post-placement optimization improves AP applicability to 100%.

Design	Utilization	Flow (%)	# EPE (Gate)	# EPE (Poly)	GDSII size (MB)	OPC Runtime (sec)	ORC Runtime (sec)
AES	70	MBOPC	16921	70601	3426	7432	731
		COPC(WO)	35413	121862	650	144	347
		COPC(V)	17958	88539	656	168	378
		COPC(HV)	17104	72659	659	192	400
	60	MBOPC	17091	73032	3416	7676	738
		COPC(WO)	39584	130383	649	144	362
		COPC(V)	17526	96594	655	168	389
		COPC(HV)	17379	75184	657	192	400
	50	MBOPC	17102	76321	3360	7834	742
		COPC(WO)	41238	137507	649	144	338
		COPC(V)	17632	103975	656	168	387
		COPC(HV)	17294	77494	659	192	401
ALU	70	MBOPC	383	28171	2943	3709	1739
		COPC(WO)	2665	64027	539	120	476
		COPC(V)	419	42216	542	136	506
		COPC(HV)	403	30817	546	160	510
	60	MBOPC	370	28520	2948	3945	1790
		COPC(WO)	2517	64989	537	120	465
		COPC(V)	410	42264	544	136	500
		COPC(HV)	401	30182	547	160	509
	50	MBOPC	365	28640	2948	4132	1740
		COPC(WO)	2692	62542	537	120	454
		COPC(V)	416	41841	543	136	503
		COPC(HV)	394	30685	547	160	508

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 90%. Poly EPE count of COPC(HV) matches that of MBOPC within 4%. For AES, COPC(HV) reduces OPC runtime over MBOPC by a factor of 40X.

Printability analysis of AP-based OPC shows that V-AP and V/H- AP can match gate poly EPE count of model-based OPC to within 4%. This is an improvement of 90%, on average, over cell-based OPC without APs. Our post-placement optimization method can achieve 100% AP applicability in designs with utilization of $<$ 70%. For designs with utilization of $>$ 70%, we can achieve up to 80% AP applicability. Our current AP insertion approach is oblivious to timing criticality of cells in the layout. We are currently incorporating timing-awareness into the AP insertion and placement perturbation methodology.

REFERENCES

1. L. Capodieci, P. Gupta, A. B. Kahng, D. Sylvester and J. Yang, "Toward a Methodology for Manufacturability Driven Design Rule Exploration", *Proc. ACM/IEEE Design Automation Conf.*, June 2004, pp. 311-316.
2. P. Gupta and F.-L. Heng, "Toward a Systematic Variation-Aware Timing Methodology", *Proc. of ACM/IEEE Design Automation Conference*, 2004, pp. 321-326.
3. T. Matsunawa, H. Nosato, H. Sakanashi, M. Murakawa, N. Murata, T. Terasawa, T. Tanaka, N. Yoshiija, O. Syga and T. Higuchi, "The novel approach for optical proximity correction using genetic algorithms", in *Proc. BACUS Symposium on Photomask Technology* Vol. 5992, 2005, pp. 54-1 - 54-9.
4. F.-L. Heng, J.-F. Lee, P. Gupta, "Toward Through-Process Layout Quality Metrics", *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing*, 2006, to appear
5. X. Wang, M. Pilloff, H. Tang and C. Wu, "Exploiting hierarchical structure to enhance cell-based RET with localized OPC reconfiguration," in *Proc. SPIE Conference on Design and Process Integration for Microelectronic Manufacturing* Vol. 5756, 2005, pp. 361-367.
6. K. Cao, S. Dobre, J. Hu, "Standard Cell Characterization Considering Lithography Induced Variations", *Proc. of ACM/IEEE Design Automation Conference*, 2006, pp. 801-804.