Electrically Driven Optical Proximity Correction Based on Linear Programming

Shayak Banerjee, Praveen Elakkumanan, Lars W. Liebmann and Michael Orshansky

1Department of Electrical and Computer Engineering, The University of Texas at Austin, TX
2IBM Corp., Hopewell Junction, NY

Abstract - Conventional optical proximity correction (OPC) tools aim to minimize edge placement errors (EPE) due to the optical and resist process by moving mask edges. However, in low-k1 lithography, especially at 45nm and beyond, printing perfect polygons is practically impossible to achieve. In addition, prohibitively high mask complexity is incurred, leading to high mask cost. Given the impossibility of perfect printing, we argue that aiming to reduce the error of electrical discrepancy between the ideal and the printed contours is a more reasonable strategy. In fact, we show that contours with non-minimal EPE may result in closer match to the desired electrical performance.

Towards achieving this objective, we developed a new electrically driven OPC (ED-OPC) algorithm. The tool combines lithography simulation with accurate electrical modeling of resist contours to predict the on/off current through a transistor gate. The computation of mask edge movements is cast as a linear program based on optical and electrical sensitivities. The objective is to minimize the error in saturation current between printed and target shapes. This optimization is then solved with fast runtime. The results on industrial 45nm SOI layouts using high-NA immersion lithography models show up to a 5% improvement in accuracy of timing over conventional OPC. This is achieved at less than 26% runtime overheads, while also lowering mask complexity by up to 43%. The results confirm that better timing accuracy can be achieved despite larger edge placement error.

I. INTRODUCTION

Semiconductor scaling has primarily been driven by advances in lithographic technologies allowing the printing of increasingly smaller feature sizes. The lack of commercially viable resist and lens materials have, thus far, prevented migration to more advanced lithographic technologies such as 154nm or EUV lithography. Consequently, Resolution Enhancement Techniques (RET) such as sub-resolution assist features (SRAF), phase-shift masking (PSM) and optical proximity correction (OPC) have become vital to ensuring high lithographic yield at the current 193nm lithography node. OPC is the technique of distorting mask features to ensure that layout features print according to specifications. The model-based OPC flow typically in use today consists of contour generation followed by optimization to reduce geometric error between the resist contour and target feature. However, recent experiments [1] have demonstrated that minimizing edge placement error does not necessarily guarantee desired electrical behavior of a particular feature.

The essential reason is that in 45nm processes, even the most extensive conventional OPC that minimizes EPE cannot produce intended polygonal shapes. This is due to several factors including diffraction-limited optics, mask errors, and variations in process conditions. Geometric mismatch leads to the inability to guarantee ideal electrical behavior, which in turn affects timing accuracy and parametric yield.

The limitations of existing OPC flows with respect to timing accuracy have been highlighted previously [1,3]. Experimental results have shown up to 36.4% increase in worst-case slack due to residual OPC errors [3]. The idea of choosing from several available OPC recipes on the basis of matching transistor currents has been explored in [1]. A technique to pass cell timing information to the OPC engine has also been proposed [2]. This is achieved by establishing a relationship between timing slack and EPE tolerance, which is subsequently used as input to a conventional OPC algorithm. While experiments are a pointer to the possibilities of tuning the OPC engine to meet design requirements, both approaches do not change the existing OPC algorithm.

Additionally, at the 45nm technology node, using OPC to print perfect polygons leads to highly complex masks which suffer from high cost and mask errors [2]. Techniques such as target smoothing [10] have been used on electrically non-critical features to reduce mask complexity by relaxing the constraints on EPE. However, such approaches have drawbacks in terms of disturbing electrical properties when used on electrically critical features.

In this paper we propose to modify the objective of OPC to minimize the electrical error, rather than edge placement error. We utilize previously developed electrical contour models [4,5,6,9] to predict timing and leakage performance of individual transistors. A rigorous linear programming formulation is then developed for computation of optimal fragment movements to compensate for the electrical error over the entire shape. This contrasts to conventional OPC, where fragments are moved independently to correct for local geometric errors. In addition to improving electrical accuracy, this technique lowers mask complexity since not all features on a mask are electrically critical. This potentially leads to lower mask costs and lower probability of mask errors. The experiments in 45nm SOI technology on polysilicon layer using high-NA immersion lithography models show up to 5% improvement in timing accuracy over...
a current state of the art OPC engine. In addition, up to 43% reduction in number of mask vertices is observed. Good convergence is obtained at less than 26% runtime overheads. The flows for the ED-OPC and a conventional EPE-based OPC are compared in Fig. 1.

![Flowchart](image-url)

**Fig. 1**: (a) Flow for conventional OPC; (b) Proposed flow for electrically driven OPC (broken lines represent changes in the flow).

Most effort in design for manufacturability (DFM) has thus far been concentrated in pushing manufacturing information upstream in the form of simulators (lithography, CMP). The proposed approach enables the communication of design information, including criticality of cell blocks, downstream. It opens up the possibility of passing timing or leakage information to the OPC engine, allowing the prioritization of either timing accuracy or leakage reduction with respect to mask costs.

The paper is organized as follows. Section II describes the contour modeling techniques used in the ED-OPC flow to predict electrical behavior. The linear-programming based strategy for fragment movement based on electrical errors is elucidated in Section III. Section IV displays timing results obtained from applying the algorithm to logic gate layouts at the polysilicon level, together with results for convergence and runtime of the proposed algorithm.

**II. ELECTRICAL SHAPE MODEL**

To enable ED-OPC, we need an accurate model to predict the electrical performance of any printed shape. We focus on polysilicon overlapping active area which represents the transistor gate and is thus essential for electrical performance. Circuit simulators, such as SPICE, implicitly assume that the gate is rectangular which allows a single value of gate length to describe the transistor. However, as described before, perfect polygons are impossible to achieve in low-k1 lithography. This calls for a model of a transistor with non-rectangular gate structure.

![Graph](image-url)

**Fig. 3**: SPICE pre-characterization of unit current as a function of gate length.

We use a previously proposed slicing technique [5] to model electrical behavior of a poly contour. The model represents a gate as a set of parallel transistors (slices). Fig. 2 shows a contour, which is sliced along the width into segments of equal width $W_{seg}$. The slice itself is assumed to be small enough to have a constant length $L(x)$. We define unit current as saturation current per unit width of a transistor. For each slice, the unit current can be computed as the unit current of a transistor with constant $L_{gate} = L(x)$ (Eq. 1):

$$I(x) = f[L(x)]$$ (1)

The dependence of this current on gate length is pre-characterized via circuit simulation (Fig. 3) to speed up evaluation. The total shape current is then given by:

$$I_{shape} = \sum x I(x)W_{seg}$$ (2)

We also account for the fact that the current density across the width of the transistor is not uniform [6],
primarily due to the non-uniformity of threshold voltage as a result of the reverse narrow width effect (RNWE). One of the main sources of threshold voltage ($V_T$) non-uniformity in shallow-trench isolation (STI) processes is the extension of the gate over the isolation area. This leads to fringing capacitances between the gate, STI sidewall and active area, which lowers $V_T$ [11]. We use 45nm SPICE models calibrated to capture RNWE. A series of SPICE simulations are performed to characterize the unit current as a function of width. The derivative of this function [9] gives the dependence of unit current on location $[e(x)]$. Fig. 4 shows this dependence. A piecewise linear model is used to represent $e(x)$ analytically:

$$ I_0 \left( \frac{I_g - I_t}{w_c} \right) x \quad 0 < x \leq w_c $$

$$ e(x) = I_0 \left( \frac{W - w_c}{w_c} \right) + \left( \frac{I_g - I_t}{w_c} \right) x \quad w_c < x \leq W - w_c $$

$$ I_0 \left( \frac{W - w_c}{w_c} \right) + \left( \frac{I_g - I_t}{w_c} \right) x \quad W - w_c < x \leq W $$

Here $w_c$, $I_0$ and $I_g$ are technology-specific fitting parameters while $W$ is the transistor width. A convenient way to account for different impact of slices at various locations is to use an edge effect factor $w(x)$:

$$ w(x) = \frac{1}{I_0} \int_{0}^{x} e(x) dx $$

The piecewise linear model of $e(x)$ allows analytical computation of the above integral. The total current for the contour is now:

$$ I_{shape} = \sum_{x} I(x)w(x) $$

Finally, the equivalent (effective) gate length can now be extracted based on the total transistor current (Eq. 6)

$$ L_{eff} = f^{-1}(I_{shape} / W) $$

This effective gate length can be used in SPICE to estimate transistor timing behavior. Specifically, we utilize it to compare timing numbers for ED-OPC against a conventional approach. While the above model has been used to describe timing behavior of a transistor, it can be used to calculate transistor leakage as well, by using similar models for off-current in place of saturation current.

III. ALGORITHM FOR EDGE MOVEMENT

The key difference between a conventional OPC and ED-OPC is the way fragment movements are determined. Fragmentation is the process of generating movable mask edges to control the shape of the contour. Conventional OPC computes edge placement error at a number of measurement locations called sites, and moves edges to minimize total EPE across all sites. Matrix OPC is one commonly used EPE-based technique for PSM and complex illumination schemes [8]. In this approach, fragment movement is done on a per-site basis and is based on mask enhancement factor (MEEF). MEEF is defined as the sensitivity of edge placement error at a particular site to fragment movement at the same or neighboring sites, and can be expressed as:

$$ MEEF_i = \frac{\partial e_i}{\partial d_j} $$

The edge placement error $e_i$ at a given site $i$ is a complex function of the mask. Given a set of $n$ fragments, a first-order Taylor series expansion is used to determine $e_i$ as a function of mask edge movements ($\Delta d_j, 1 \leq j \leq n$) [8].

$$ e_i = f(d_j) \quad 1 \leq j \leq n $$

$$ e_i = e_{io} + \sum_{j=1}^{n} \frac{\partial e_i}{\partial d_j} \Delta d_j $$

The partial derivative ($\partial e_i / \partial d_j$) is computed numerically from the Hopkins partial coherence equations [7, 8]. For ease of computation, it can be assumed that the EPE at a site is primarily affected by the movement of the fragment at the same site i.e. self-MEEF is dominant. This simplifies Eq. 8 to include just one partial derivative term.

$$ e_i = e_{io} + \frac{\partial e_i}{\partial d_i} \Delta d_i $$

The mask edge movement at a given site is then determined by setting EPE to zero which computes $\Delta d_i$ as:

$$ \Delta d_i = - \frac{e_{io}}{\frac{\partial e_i}{\partial d_i}} $$

The algorithm for conventional OPC is summarized as:

**Algorithm:** Conventional Optical Proximity Correction

**Inputs:**
- $L$ – a drawn cell layout
- $F$ – set of mask fragments
- $E$ – set of edge placement errors corresponding to each fragment

**Output:** $L'$ – proximity corrected layout
1. Initialize $D \leftarrow 0$ (set of mask edge movements)
2. for $j \leftarrow 1$ to $iter$
3. foreach $f_i \in F$
4. compute $d_i \in D$ (Eq. 10)
5. end
6. $L' \leftarrow \text{Move}_\text{Edges}(D)$
7. update $E \leftarrow \text{Litho}_\text{Simulation}(L')$
8. next $j$
9. Output $L'$

Although the above approach allows fast computation for a large number of fragments, it has three main drawbacks. Firstly, the objective of conventional OPC is minimization of EPE, which as discussed earlier, can lead to a large electrical error and a larger mask cost. Secondly, the result of correcting for EPE on a per-site basis provides sub-optimal results because the interaction between sites is not captured. While this issue is addressed at a higher computational cost [8], specifically in the case of electrical matching the electrical metric depends on the entire shape and not just one particular fragment. This means for the same EPE a better solution may be found if a global electrical metric is optimized. Thirdly, a large amount of effort is spent on electrically non-critical features, which may not require a tight degree of control. The electrically driven formulation for edge movement based on linear programming that we advance solves these problems.

Our premise in formulating ED-OPC is that increasing programming that we advance solves these problems.

Consider a pair of parallel mask edges of length $w_{\text{frag}}$ as shown in Fig. 5. Henceforth, we will use the term electrical fragment to describe such a structure. Given a set of $n$ shapes, each with $m_i$ electrical fragments ($1 \leq i \leq n$), we can further express ESE for each shape in terms of individual electrical fragment currents as:

$$ESE_i = \sum_{i=1}^{m_i} ESE_{i,i}$$

where $ESE_{i,i}$ is the difference in saturation currents of the contour and target shape ($I_{\text{contour}}$ and $I_{\text{target}}$, respectively). For this purpose, we define the electrical shape error as:

$$ESE_i = \left| I_{\text{contour}} - I_{\text{target}} \right|$$

where $n$ is the number of transistors in the layout. It is also important to ensure that leakage current is not made worse by ED-OPC since in this case EPE is not directly minimized.

The dependence of leakage on gate length is exponential. In allowing a non-rectangular contour, we enable the possibility of producing regions of narrow gate length. Such regions will suffer from very high leakage, increasing the leakage of the overall transistor. Hence, we formulate electrically driven OPC as the algorithm to minimize the total electrical shape error for all shapes, while ensuring leakage is not significantly increased:

$$\begin{array}{l}
\text{min} \quad \sum_{i=1}^{n} ESE_i \\
\text{s.t.} \quad I_{\text{off}} \leq I_{\text{off,max}} \quad 1 \leq i \leq n
\end{array}$$

A similar expression is used to determine the change in the electrical fragment off-current due to edge movements:

$$\Delta I_{\text{off,frag}} = w_{\text{frag}} \left( \frac{\partial e_{\text{off}}}{\partial d_{\text{left}}} \Delta d_{\text{left}} + \frac{\partial e_{\text{off}}}{\partial d_{\text{right}}} \Delta d_{\text{right}} \right)$$

Using Eq. 16, the change in ESE due to mask edge movements can be rewritten as:

$$ESE_i = \left| I_{\text{contour}} - I_{\text{target}} \right|$$

$$\left| I_{\text{contour}} - I_{\text{target}} \right| = \left| I_{\text{contour}} - I_{\text{target}} \right|$$

For a given electrical fragment, a first-order Taylor series expansion is used to predict change in the on-current as a function of mask edge movements:

$$\Delta I_{\text{on,frag}} = w_{\text{frag}} \left( \frac{\partial e_{\text{on}}}{\partial d_{\text{left}}} \Delta d_{\text{left}} + \frac{\partial e_{\text{on}}}{\partial d_{\text{right}}} \Delta d_{\text{right}} \right)$$

Similar to Eq. 9, we assume that electrical fragment current is primarily affected by the movement of the two edges abutting the fragment. The electrical fragment width ($w_{\text{frag}}$) is factorized out of this expression and the partial derivatives are written in terms of current per unit length ($i_{\text{on}}$). This allows incorporation of edge effects into the term $w_{\text{frag}}$ using the model described in section II.

We refer to the partial derivative ($\partial i_{\text{on}}/\partial d$) as electrical MEEF (E-MEEF). Applying chain rule, we can write the electrical MEEF as:

$$E \cdot \text{MEEF} = \frac{\partial i_{\text{on}}}{\partial d} = \left( \frac{\partial e_{\text{on}}}{\partial d} \right) \left( \frac{\partial d}{\partial d} \right) = \left( \frac{\partial e_{\text{on}}}{\partial L} \right) \left( \frac{\partial d}{\partial d} \right)$$

Eq. 15 describes E-MEEF as a product of electrical ($\partial i_{\text{on}}/\partial d$) and optical ($\partial e_{\text{on}}/\partial d$) sensitivities. Electrical sensitivities are obtained from SPICE pre-characterized dependence of current on gate length, while optical sensitivity is the MEEF. We now rewrite Eq. 14 as:

$$\Delta I_{\text{off,frag}} = w_{\text{frag}} \frac{\partial e_{\text{off}}}{\partial d} \Delta d_{\text{left}} + \frac{\partial e_{\text{off}}}{\partial d} \Delta d_{\text{right}}$$

A similar expression is used to determine the change in the electrical fragment off-current due to edge movements:

$$\Delta I_{\text{off,frag}} = w_{\text{frag}} \frac{\partial e_{\text{off}}}{\partial d} \Delta d_{\text{left}} + \frac{\partial e_{\text{off}}}{\partial d} \Delta d_{\text{right}}$$

Using Eq. 16, the change in ESE due to mask edge movements can be rewritten as:

$$ESE_i = \left| I_{\text{contour}} - I_{\text{target}} \right|$$

$$\left| I_{\text{contour}} - I_{\text{target}} \right| = \left| I_{\text{contour}} - I_{\text{target}} \right|$$
II. The assumption of linearity of the functions are obtained using the contour models described in Section I efficiently. The electrical fragment currents iteration of the OPC algorithm as an LP allows it to be done that minimize electrical error. The ability to solve a single

Finally, using equations 17 and 18, we rewrite Eq. 12 as:

\[
\min \sum_{i} \left[ \sum_{j} I_{on,frag,ij} \Delta d_{i,j} \right] - \sum_{i} \left[ \sum_{j} I_{off,frag,ij} \Delta d_{i,j} \right]
\]

\[
s.t. \sum_{i} \left[ \sum_{j} I_{on,frag,ij} \Delta d_{i,j} \right] \leq I_{on,tgt,i}
\]

\[
\sum_{i} \left[ \sum_{j} I_{off,frag,ij} \Delta d_{i,j} \right] \leq I_{off,max,i}
\]

\[
-\eta \Delta d_{i,j} \leq \Delta d_{i,j} \eta \quad \forall i, j
\]

(19)

The algorithm for ED-OPC is summarized as:

**Algorithm: ED-OPC**

**Inputs:**
- \( L \) – a drawn cell layout
- \( F \) – set of electrical fragments
- \( S \) – set of EPE sensitivities corresponding to each electrical fragment
- \( L_{on} \) – set of electrical fragment on-currents
- \( L_{off} \) – set of electrical fragment on-currents
- \( iter \) – number of iterations

**Output:**
- \( L' \) – proximity corrected layout
  1. Initialize \( D \leftarrow 0 \) (set of mask edge movements)
  2. for \( j \leftarrow 1 \) to \( iter \)
  3. \( D \leftarrow \text{LP Solve}(S', I_{on}, I_{off}) \) (Eq. 19)
  4. \( L' \leftarrow \text{Move Edges}(D) \)
  5. update \( S', I_{on}, I_{off} \) ← \( \text{Litho Simulation}(L') \)
  6. next \( j \)
  7. **Output** \( L' \)

The above formulation leads to a linear program (LP) which can be solved for a set of mask edge movements \( \Delta d_i \) that minimize electrical error. The ability to solve a single iteration of the OPC algorithm as an LP allows it to be done efficiently. The electrical fragment currents \( I_{on,frag,ij} \) and \( I_{off,frag,ij} \) are obtained using the contour models described in Section II. The assumption of linearity of the functions \( I_{on,frag} \) and \( I_{off,frag} \) in equations 14 and 17 is only valid over a small range. We limit this range by imposing constraints \( \eta \) on the magnitude of allowed mask edge movements \( \Delta d_{i,j} \).

This helps reduce modeling error-related inaccuracies and oscillations in the solution. In order to ensure that geometric pattern fidelity is not catastrophically jeopardized, a constraint on the maximum EPE can be added in the formulation. This will act to prevent potential catastrophic yield and overlay problems.

IV. Results

The algorithm for ED-OPC was implemented using the Calibre Workbench suite of tools with high-NA immersion lithography optical and resist models. The Matrix-OPC functionality available in Calibre [8] was used to generate self-MEEF values for electrical fragments. GNU Linear Programming Kit (version 4.20) [12] was used to solve the linear program using primal-dual interior point algorithms. All electrical fragment processing was performed by converting contour and OPC layers to text-form, which was then handled using Perl scripts. SRAF insertion was done according to conventional rules. Experiments were performed on logic cells in 45nm technology using a constant size fragmentation scheme over gate regions.

Table I shows post-OPC cell timing numbers. Delay values are normalized to unit inverter delay. We generate worst-case \( (T_{on}) \) and best-case \( (T_{off}) \) delays for each cell using SPICE, assuming all gate lengths at nominal. We then perform OPC on the cell layout, generate the final contour, extract the effective gate length \( (L_{eff}) \) and use this in a SPICE simulation to determine cell delays after OPC. This is done for both conventional and electrically-driven OPC. Table I shows that while a conventional OPC provides timing accuracy to within 2.04% on an average, ED-OPC improves the timing accuracy to within 0.44%.

We further computed leakage numbers for best timing match for each cell. This is done similar to above, but by calculating the effective gate length for leakage \( (L_{eff,leak}) \) in

### TABLE I

**ELECTRICAL (TIMING) ACCURACY ACHIEVED BY CONVENTIONAL OPC AND ED-OPC**

<table>
<thead>
<tr>
<th>Cell</th>
<th>Transistor Count</th>
<th>Nominal (SPICE) ( T_{on} / T_{off} )</th>
<th>Conventional OPC ( T_{on} / T_{off} )</th>
<th>Timing Error (Conv. OPC) (%)</th>
<th>ED-OPC ( T_{on} / T_{off} )</th>
<th>Timing Error (ED-OPC) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV</td>
<td>2</td>
<td>1.06 / 1.00</td>
<td>1.10 / 1.00</td>
<td>-0.68 / 2.84</td>
<td>1.07 / 0.99</td>
<td>0.59 / -1.24</td>
</tr>
<tr>
<td>BUF</td>
<td>4</td>
<td>1.10 / 1.09</td>
<td>1.11 / 1.11</td>
<td>0.90 / 1.92</td>
<td>1.09 / 1.09</td>
<td>-0.96 / 0.58</td>
</tr>
<tr>
<td>NAND2</td>
<td>4</td>
<td>1.48 / 0.79</td>
<td>1.48 / 0.83</td>
<td>-0.52 / 5.16</td>
<td>1.49 / 0.80</td>
<td>0.24 / 0.75</td>
</tr>
<tr>
<td>NOR3</td>
<td>6</td>
<td>1.80 / 0.72</td>
<td>1.78 / 0.74</td>
<td>-1.25 / 2.50</td>
<td>1.79 / 0.72</td>
<td>-0.30 / -0.22</td>
</tr>
<tr>
<td>AND2</td>
<td>6</td>
<td>1.54 / 1.07</td>
<td>1.62 / 1.11</td>
<td>4.75 / 3.23</td>
<td>1.55 / 1.07</td>
<td>0.16 / 0.02</td>
</tr>
<tr>
<td>AND3</td>
<td>8</td>
<td>1.76 / 0.99</td>
<td>1.76 / 1.02</td>
<td>0.18 / 2.45</td>
<td>1.75 / 1.00</td>
<td>-0.22 / 0.96</td>
</tr>
<tr>
<td>AND4</td>
<td>10</td>
<td>1.98 / 0.95</td>
<td>2.01 / 0.96</td>
<td>1.56 / 0.60</td>
<td>1.98 / 0.95</td>
<td>0.07 / -0.69</td>
</tr>
<tr>
<td>XNOR2</td>
<td>10</td>
<td>1.51 / 1.28</td>
<td>1.54 / 1.28</td>
<td>1.75 / 0.17</td>
<td>1.50 / 1.28</td>
<td>-0.53 / 0.13</td>
</tr>
<tr>
<td>XOR</td>
<td>11</td>
<td>1.54 / 1.26</td>
<td>1.54 / 1.28</td>
<td>-0.15 / 1.71</td>
<td>1.53 / 1.26</td>
<td>-0.69 / 0.32</td>
</tr>
<tr>
<td>Half ADDER</td>
<td>14</td>
<td>1.83 / 1.15</td>
<td>1.95 / 1.18</td>
<td>6.60 / 2.23</td>
<td>1.84 / 1.15</td>
<td>0.47 / -0.38</td>
</tr>
<tr>
<td>Full ADDER</td>
<td>28</td>
<td>1.18 / 0.96</td>
<td>1.19 / 0.98</td>
<td>1.05 / 1.97</td>
<td>1.18 / 0.97</td>
<td>0.12 / 0.13</td>
</tr>
</tbody>
</table>

**Average Absolute Error**

2.04 0.44
place of $L_{\text{eff}}$. Results show 8.6% average increase in pull-up ($P_{\text{leak}}$) and pull-down ($N_{\text{leak}}$) leakage for the best timing match (Table II). Even though some individual cells exhibit higher leakage, if the mix of cells in the design is uniform, the average leakage is an appropriate metric for comparison.

We evaluated runtime overhead of ED-OPC due to the additional contour modeling and LP-based edge movement computation. To enable a fair comparison, we implemented a text-based version of conventional OPC using the algorithm described in Section III. Simulations were run on a 400MHz IBM PowerPC POWER3 processor with 512KB L1 cache. Results (Table III) indicate that runtime increase due to ED-OPC is 18% on an average. One of the main reasons why runtime degradation is not high is because we achieve high electrical accuracy with fewer fragments in comparison to a conventional OPC. We further calculated the complexity of mask shapes produced by both conventional and ED-OPC. This was done by extracting the transistor gate regions from the OPC layer and counting the number of vertices. The conventional OPC uses an adaptive fragmentation scheme with a minimum fragment size of 24nm. Adjustable fragmentation schemes could be used for ED-OPC, but we found that constant fragmentation with a 40nm fragment width was adequate. Up to 43% reduction in mask complexity was achieved by ED-OPC (Table III).

In order to observe convergence of the ED-OPC algorithm we measured the electrical error after each OPC iteration. This was done by extracting the effective gate lengths at each step and running a SPICE simulation to generate delays. The results are shown in Fig. 6, where the timing error is plotted as a function of the iteration number for the first six iterations. It is seen that the timing error quickly converges to within 2% in the specified number of iterations for all test cases.

To highlight the difference in resist contours produced by the two OPC techniques, we compared EPE histograms for the same layout (Fig. 7).

![Fig. 6: Convergence of ED-OPC: timing error plotted for first 6 iterations.](image)

### TABLE II

**Leakage Degradation at OPC Solution with Best Timing Match**

<table>
<thead>
<tr>
<th>Cell</th>
<th>Ideal (SPICE) $(N_{\text{leak}} / P_{\text{leak}})$</th>
<th>Conv. OPC $(N_{\text{leak}} / P_{\text{leak}})$</th>
<th>Leakage Degradation (Conv. OPC) (%)</th>
<th>ED-OPC $(N_{\text{leak}} / P_{\text{leak}})$</th>
<th>Leakage Degradation (ED-OPC) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV</td>
<td>1.31 / 1.00</td>
<td>1.02 / 1.09</td>
<td>-22.2 / 8.6</td>
<td>1.47 / 1.08</td>
<td>12.7 / 8.1</td>
</tr>
<tr>
<td>BUF</td>
<td>2.12 / 2.16</td>
<td>1.69 / 1.97</td>
<td>-20.1 / -9.2</td>
<td>2.35 / 2.56</td>
<td>10.6 / 18.5</td>
</tr>
<tr>
<td>NAND2</td>
<td>0.09 / 1.73</td>
<td>0.09 / 0.82</td>
<td>-4.9 / -25.6</td>
<td>0.09 / 1.83</td>
<td>-0.2 / 5.7</td>
</tr>
<tr>
<td>NOR3</td>
<td>2.96 / 0.05</td>
<td>2.36 / 0.05</td>
<td>-20.5 / 3.8</td>
<td>2.92 / 0.06</td>
<td>-1.6 / 15.0</td>
</tr>
<tr>
<td>AND2</td>
<td>1.00 / 2.34</td>
<td>0.62 / 1.78</td>
<td>-38.7 / -23.7</td>
<td>1.20 / 2.20</td>
<td>19.1 / -5.9</td>
</tr>
<tr>
<td>AND3</td>
<td>0.96 / 2.92</td>
<td>0.59 / 2.03</td>
<td>-38.6 / -30.4</td>
<td>1.14 / 2.85</td>
<td>18.5 / -2.2</td>
</tr>
<tr>
<td>AND4</td>
<td>0.95 / 3.50</td>
<td>0.58 / 2.30</td>
<td>-39.2 / -34.5</td>
<td>1.04 / 3.60</td>
<td>9.5 / 2.7</td>
</tr>
<tr>
<td>XNOR2</td>
<td>5.59 / 5.62</td>
<td>5.26 / 4.95</td>
<td>-5.9 / -11.9</td>
<td>5.51 / 5.86</td>
<td>-1.4 / 4.1</td>
</tr>
<tr>
<td>Half ADDER</td>
<td>5.69 / 6.68</td>
<td>4.16 / 5.17</td>
<td>-26.9 / -22.7</td>
<td>6.35 / 7.34</td>
<td>11.6 / 9.9</td>
</tr>
<tr>
<td>Full ADDER</td>
<td>5.94 / 6.90</td>
<td>4.59 / 5.56</td>
<td>-22.7 / -19.4</td>
<td>5.73 / 7.05</td>
<td>-3.7 / 2.1</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td></td>
<td>-20.5</td>
<td>8.6</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE III

**Fragment Count and Runtime for Conventional versus ED-OPC**

<table>
<thead>
<tr>
<th>Cell</th>
<th>Conventional OPC</th>
<th>ED-OPC</th>
<th>Runtime (s)</th>
<th>Fragment Count</th>
<th>Runtime (s)</th>
<th>Fragment Count</th>
<th>Runtime Increase (%)</th>
<th>Mask Complexity Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV</td>
<td>76</td>
<td>46</td>
<td>37.44</td>
<td>46</td>
<td>39.64</td>
<td>5.9</td>
<td>39.5</td>
<td>29.2</td>
</tr>
<tr>
<td>BUF</td>
<td>144</td>
<td>102</td>
<td>36.33</td>
<td>124</td>
<td>40.82</td>
<td>12.4</td>
<td>16.2</td>
<td>22.4</td>
</tr>
<tr>
<td>NAND2</td>
<td>148</td>
<td>124</td>
<td>35.95</td>
<td>192</td>
<td>44.35</td>
<td>23.4</td>
<td>14.3</td>
<td>16.2</td>
</tr>
<tr>
<td>NOR3</td>
<td>224</td>
<td>172</td>
<td>35.54</td>
<td>216</td>
<td>44.25</td>
<td>24.5</td>
<td>14.3</td>
<td>16.2</td>
</tr>
<tr>
<td>AND2</td>
<td>160</td>
<td>172</td>
<td>41.72</td>
<td>57</td>
<td>45.93</td>
<td>10.1</td>
<td>30.0</td>
<td>16.2</td>
</tr>
<tr>
<td>AND3</td>
<td>222</td>
<td>156</td>
<td>36.83</td>
<td>212</td>
<td>44.11</td>
<td>19.8</td>
<td>29.7</td>
<td>16.2</td>
</tr>
<tr>
<td>AND4</td>
<td>276</td>
<td>212</td>
<td>67.57</td>
<td>212</td>
<td>76.07</td>
<td>12.6</td>
<td>23.2</td>
<td>16.2</td>
</tr>
<tr>
<td>XNOR2</td>
<td>380</td>
<td>304</td>
<td>59.72</td>
<td>304</td>
<td>71.49</td>
<td>19.7</td>
<td>20.0</td>
<td>16.2</td>
</tr>
<tr>
<td>XOR2</td>
<td>336</td>
<td>190</td>
<td>40.29</td>
<td>190</td>
<td>50.67</td>
<td>25.8</td>
<td>43.5</td>
<td>16.2</td>
</tr>
<tr>
<td>Half ADDER</td>
<td>624</td>
<td>430</td>
<td>66.85</td>
<td>79.91</td>
<td>19.5</td>
<td>31.1</td>
<td>16.2</td>
<td>16.2</td>
</tr>
<tr>
<td>Full ADDER</td>
<td>1326</td>
<td>796</td>
<td>115.11</td>
<td>140.45</td>
<td>22.0</td>
<td>40.0</td>
<td>16.2</td>
<td>16.2</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17.8</td>
<td>28.8</td>
</tr>
</tbody>
</table>
IV. CONCLUSION

We have presented a method for optical proximity correction which uses current-matching, as opposed to edge placement errors, as an objective to improve timing accuracy. The method is based on contour generation coupled with electrical modeling to predict saturation and leakage currents. This is followed by a linear-program based calculation of mask edge movements to compensate for the difference in contour and target currents. Results show up to 5% improvement in timing accuracy using LP-based ED-OPC. Low runtime overheads have been demonstrated along with convergence to within 2% timing accuracy in a small number of iterations. Comparison with a conventional OPC also shows that up to 43% reduction in the number of mask vertices can be achieved. Higher EPE for ED-OPC contours show that while conventional OPC does a good job of shape matching, it falls short in terms of electrical performance.

The benefits of an increase in timing accuracy can be observed in the form of reduced guard-banding for optical effects on the design side, leading to less pessimistic designs. Also, reduction in mask complexity leads to lower mask costs and mask error probabilities. Electrically driven optical proximity correction allows the release of design information further downstream in the manufacturing flow, closing an otherwise open DFM loop.

ACKNOWLEDGMENTS

This research was supported by grants from SRC, C2S2, NSF and the IBM Ph.D. Fellowship.

REFERENCES