# A Fast Manufacturability Aware Optical Proximity Correction (OPC) Algorithm with Adaptive Wafer Image Estimation

Ahmed Awad, Atsushi Takahashi
Tokyo Institute of Technology, Tokyo 152–8550
Email: {awad.ahmed, atsushi}@eda.ce.titech.ac.jp

Chikaaki Kodama
Toshiba Corporation, Kanagawa 247–8585
Email: chikaaki1.kodama@toshiba.co.jp

***Abstract*—Aggressive Optical Proximity Correction (OPC) has been widely adopted in optical lithography to preserve circuit performance for sub-20nm technology nodes. However, complex mask patterns are output resulting in lower mask manufacturability and large computational time. In this paper, we propose a fast OPC algorithm in which intensity estimation during OPC is improved for better pattern fidelity and in which post processing to effectively improve mask manufacturability with preserving acceptable pattern fidelity is executed.**

## I. Introduction

Optical micro-lithography forms the heart of Integrated Circuit (IC) fabrication process. However, as advanced technology nodes continue scaling down into sub-20nm regime, wafer image quality degradation is expected (as illustrated in Fig.1) since the industry is stuck at 193nm immersion lithography due to its practical limit. Furthermore, Next Generation Lithography (NGL) is still not in the track. Therefore, the industry relies heavily on Optical Proximity Correction (OPC) to improve pattern fidelity in the foreseeable future [1][2].

During OPC, mask pattern is iteratively adjusted followed by wafer image simulation to guide the next OPC response till sufficient image quality (pattern fidelity) is obtained [3].

Wafer image estimation plays a key factor in guiding OPC algorithm response and consequently in final wafer image quality. Image estimation accuracy is directly related to light intensity estimation in each pixel by the mask. Intensity difference map was recently introduced to accelerate intensity estimation with sufficient accuracy [4]. This approach estimates intensity well since intensity difference map is almost invariant of mask layout changes. However, this estimation needs to be improved around the target boundaries.

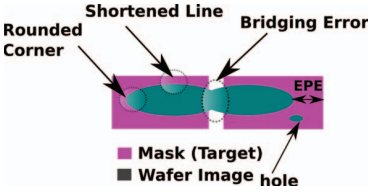



Fig. 1: Wafer image distortions.

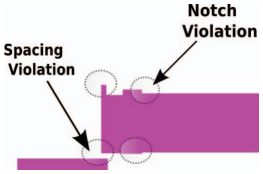



Fig. 2: Design rules violations.

To keep pace with advanced technology nodes, high level of aggressiveness in OPC algorithms is required to guarantee pattern fidelity. Consequently, complex mask layouts are outputted resulting in lower mask manufacturability. To improve mask manufacturability, minimum allowable mask notch and minimum spacing between patterns mask design rules are provided to be considered either in pre-processing or to be included in the OPC recipe [5]. However, considering such rules during OPC abruptly increases computational time for advanced technology nodes since low stability and low convergence are expected compared to relaxed rules. Fig.2 illustrates examples of design rules violations. In this paper, an adaptive intensity difference map is proposed to improve the accuracy of wafer image estimation with less computational time overhead. Furthermore, a fast manufacturability aware post processing algorithm is proposed. In our proposed OPC algorithm, a mask without considering manufacturability is obtained in short time by using improved image estimation. Then, manufacturability is improved by applying a fast post processing in which improved image estimation is exploited to keep acceptable wafer image quality. Our contributions are summarized as follows:

- A new algorithm is proposed to iteratively adapt intensity difference map following mask changes for more accurate intensity estimation and thus, better wafer image estimation. This is achieved by exploiting the observed relation between intensity difference map and local mask density.
- A fast post processing algorithm is proposed to improve mask manufacturability with keeping pattern fidelity. This is achieved by features concatenation and shifting following a fast Edge Placement Error (EPE) prediction model with exploiting intensity difference map adaptation.

The rest of this paper is organized as following: Previous work is presented in section II. Section III proposes lithographic terminology and evaluation parameters. Algorithm overview is proposed in Section IV. Adaptive intensity difference map is proposed in section V and mask manufacturability aware model is presented in section VI. Detailed algorithm is presented in Section VII. Experimental results are analyzed in section VIII and section IX concludes this paper.

## II. Previous Work

Several wafer image estimation algorithms have been proposed in the literature to OPC with pattern fidelity preservation such as Mask Optimization Solution With Process Window Aware Inverse Correction (MOSAIC), linearized models for image computation, and intensity difference map [6][7][8][4]. However, this estimation needs adaptive improvement to keep pace with advanced technology nodes.

Previously published OPC algorithms have good results in terms of wafer image quality such as Inverse Lithography Technology (ILT) algorithms [9][10] [11]. However, complex mask patterns are outputted resulting in lower mask manufacturability. To tackle this factor, design aware OPC has been proposed in [12] in which intensive analysis is applied prior

This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (B)25280013.

to OPC to define less critical regions on which less aggressive OPC is applied. Additionally, Restricted Design Rules (RDRs) are applied during OPC algorithm as in [5][13]. Hotspots elimination in preprocessing for better manufacturability was proposed in [14]. An effective post-stage optimization in which mask notch rule is satisfied by exploiting keep out zones to guide OPC was published in [15]. However, a generic post processing algorithm is required to improve manufacturability of a generic OPCed mask for advanced technology nodes.

## III. Preliminaries and Evaluation Parameters

### A. Problem Description

Given a target pattern defined in region of pixels, the objective is to find a mask solution in the same region with acceptable wafer image quality and high mask manufacturability within a short computational time.

### B. Lithographic Terminology

Let $R$ be a region of pixels in which target pattern $T$ and mask $M$ are defined where $T \subset R$ and $M \subset R$. $T$ and $M$ consist of a set of rectilinear non-overlapped polygons where a polygon consists of a set of connected pixels. An edge is a horizontal or vertical line connecting two corners of a polygon boundary. Let $l(e)$ denote the length of an edge $e$.

Let $E_T = \{e_0, e_1, \ldots, e_{N_T}\}$ denote the set of edges defined on the boundary of polygons in target $T$. Fig.3-a illustrates a target $T$ whose boundary consists of 6 edges.

In this paper, a polygon in a mask is either a core-polygon, serif, or a Sub-Resolution Assisting Feature (SRAF) as illustrated in Fig.3-b. A core-polygon corresponds to a polygon in the target. It is obtained by adding/removing rectangles on/from parts of the boundary of a polygon in $T$. The edge in the boundary of a polygon in $T$ which overlaps with a rectangle edge in a core-polygon in $M$ is called a reference edge to all the edges of that rectangle in the mask boundary. For example, in Fig.3, edge $e_0$ in the boundary of $T$ is reference edge to edges $e'_0, e'_1, e'_2, \ldots e_m$ in the mask boundary. $Ref(e')$ denotes the reference edge in $T$ for edge $e'$ in $M$. A serif is square that is on target polygon corner. An SRAF is a long bar which is parallel to target polygon edges.

Let $E_M = E_{\rm r} \cap E_{\rm nr}$ be the set of edges in mask $M$ boundaries where $E_{\rm r}$ and $E_{\rm nr}$ denote the set of edges that have reference edges (on core-polygon boundaries) and those that do not have (on SRAF/serif boundaries), respectively.

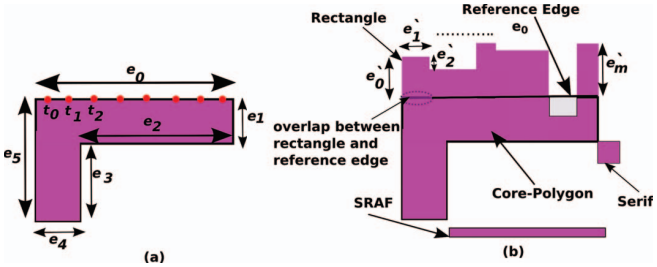



Fig. 3: (a) Target Pattern $T$. (b) Mask $M$ of the target $T$.

### C. Lithographic Model

Let $I(M)$ and $G(M)$ be the intensity map and the wafer image by mask $M$, respectively. Sum of Coherent Systems (SOCS) model is often used in OPC algorithms to obtain intensity map [16]. In this model, the optical system is decomposed into a set of coherent kernels. Each kernel has an eigenfunction that represents its filtering behavior and eigenvalue representing its contribution to the intensity value. A mask $M$ is convoluted with each kernel eigenfunction given shown in eq.(1) where $\sigma_k$ and $\phi_k$ represents the eigenvalue and the eigenfunction for kernel $k$, respectively, $K$ is the set of kernels used to obtain intensity map $I(M)$, and $\otimes$ denotes convolution operation.

$$I(M,K) = \sum_{k \in K} \sigma_k |(\phi_k \otimes M)|^2 \quad (1)$$

Constant Threshold Resist (CTR) model is often used to obtain wafer image. Intensity threshold $I_{\rm th}$ is predefined and $G(M)$ is defined as the set of pixels whose intensities are greater than or equal to $I_{\rm th}$, i.e, $G(M) = \{p \in R | I(M,p) \geq I_{\rm th}\}$ where $I(M,p)$ is the intensity in pixel $p$ using mask $M$.

### D. Wafer Image Quality Evaluation

Tap points are defined on target boundary edges. For example, in Fig.3-a tap points on the upper edge are shown.

Edge Placement Error (EPE) of a point $t$, denoted by $epe(T, G(M), t)$, is the geometrical distance between point $t$ on $T$ and its corresponding point on $G(M)$ as shown in Fig.1. EPE evaluation can be relaxed as long as no electric violations. Thus, we define $d_{\rm epe}$ as maximum allowable EPE. A tap point $t$ is considered in EPE state if $epe(T, G(M), t) > d_{\rm epe}$. In this paper, we use $\#EPEV(M)$, which is the number of tap points in EPE state, as wafer image quality (pattern fidelity) evaluation metric of a mask $M$.

### E. Mask Manufacturability Evaluation

Mask notch rule defines $d_{\rm notch}$ which is the minimum allowable edge length in the mask. Mask notch rule says: $\forall e' \in E_M$, $l(e') \geq d_{\rm notch}$. Thus, the number of mask notch rule violations in mask $M$, denoted by $\#NotchV(M)$, is given in eq.(2).

$$\#NotchV(M) = |\{e' \mid e' \in E_M, l(e') < d_{\rm notch}\}|. \quad (2)$$

Spacing rule defines $d_{\rm min}$ as the minimum allowable distance between patterns in the mask. Let $D(e'_i, e'_j)$ denotes Manhattan distance between pair of edges $(e'_i, e'_j)$ in $M$. Spacing rule says that $D(e'_i, e'_j) \geq d_{\rm min}$ where $(e'_i, e'_j)$ is called a comparison pair which can be obtained by bounding techniques. Two edges in the mask are said to be a comparison pair if they belong to different polygons, or they belong to the same core-polygon without having intersection between their reference edges. Thus, the number of spacing rule violations of mask $M$, denoted by $\#SpaceV(M)$, is given in eq.(3) where $C_{\rm p}$ is the set of comparison pairs in $M$.

$$\#SpaceV(M) = |\{(e'_i, e'_j) \mid (e'_i, e'_j) \in C_{\rm p}, D(e'_i, e'_j) < d_{\rm min}\}|. \quad (3)$$

In this paper, mask manufacturability is evaluated in terms of $\#NotchV(M)$ and $\#SpaceV(M)$ in addition to the mask data volume $V(M)$. Minimizing these parameters means higher mask manufacturability, and thus, lower mask manufacturing cost.

## IV. Algorithm Overview

Our OPC algorithm consists of three modules as shown in Fig.4: Initialization, EPE optimization, and Post OPC modules.

In initialization module, two intensity maps $I_2$ and $I_3$ are simulated from target $T$. One OPC step is applied on $T$ to generate first mask $M_1$ from which intensity maps $I_2$ and $I_3$ are re-simulated to derive the correction coefficients.

In EPE optimization module, intensity maps $I_1, I_2, I_3$ are

used to derive a modified mask. $I_1$ is generated for each mask $M$ by lithography simulation. $I_2$ and $I_3$ (generated in the initialization module) are used as difference map to compensate $I_1$. In each OPC step, $I_2$ and $I_3$ are adapted by using the derived correction coefficients during initialization.

Post OPC module, whose purpose is to improve mask manufacturability, consists of 3 main steps as illustrated in Fig.5: Mask notch violations minimization, spacing violations minimization, and mask data volume reduction.

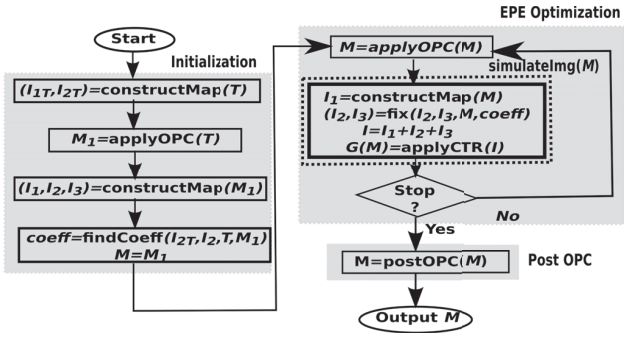



Fig. 4: Proposed Algorithm

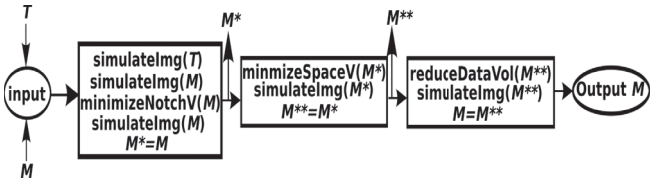



Fig. 5: Flowchart of Post-OPC Algorithm

## V. Adaptive Intensity Difference Map Modeling

### A. Intensity Difference Map Definition

Intensity difference map is the difference between two intensity maps obtained in the same region of pixels using different sets of kernels. Let $K$ and $K'$ be two sets of kernels, intensity difference map of a mask $M$ between $K$ and $K'$, denoted by $I_{\text{diff}}(M, K, K')$ is given in eq(4). $I_{\text{diff}}(M, K, K', p)$ represents intensity difference map value in pixel $p$.

$$I_{\text{diff}}(M, K, K') = I(M, K) - I(M, K') \tag{4}$$

### B. Drawbacks of Traditional Intensity Difference Map Usage

One of intensity difference map usages is the algorithm published in [4]. In this algorithm, from a target pattern $T$, $I_{\text{diff}}(T, K, \{k_0\})$ is obtained during initialization where $K$ is the set of SOCS kernels and $k_0 \in K$ is the top weight kernel. $I_{\text{diff}}$ works then as a compensative map to the obtained intensity map of a mask $M$ using $k_0$ as given in eq.(5) to estimate the intensity map $I(M)$ with only one convolution.

$$I(M) \approx I(M, \{k_0\}) + I_{\text{diff}}(T, K, \{k_0\}) \tag{5}$$

However, lack of accuracy in the information that $I_{\text{diff}}$ entails around target boundaries is expected. The reason is that, $I_{\text{diff}}$ is constructed from $T$ whose polygon edges locations are typically far from the final mask solution with high frequencies located around target boundaries when Fast Fourier Transform (FFT) is applied. Consequently, those frequencies are suppressed by low pass filters (kernels). Furthermore, this difference map usage assumes that $I_{\text{diff}}$ is invariant of mask shape slight changes while dramatic changes in mask shape are possible. Therefore, $I_{\text{diff}}$ needs to be adaptively corrected following mask changes.

### C. Intensity Difference Map Adaptation Model

To improve the accuracy of intensity estimation, lower weight kernels effect against mask layout is examined, and a linearity of that effect against local mask density is observed. Even though the deviation of the coefficient of linearity is found small in local region, the coefficient of linearity depends on local mask structure and it changes globally. It is not easy to determine the coefficient by characterizing local mask structure. Therefore, in our work, the coefficient is estimated by comparing two intensity maps in early stage of OPC.

Let $r(p) \subset R$ be a region of pixels whose center is pixel $p$ as shown in Fig.6-a. Local mask density of pixel $p$, denoted by $\delta(M, p)$, is the number of mask pixels intersecting region $r(p)$, i.e., $\delta(M, p) = |M \cap r(p)|$. Let $K$ be a set of kernels whose top weight kernel is $k_0 \in K$. Let $M_{\text{ref}}$ and $M$ be two masks of a target $T$. With considering the linearity observation of lower weight kernels against local mask density, the difference map of mask $M$ in pixel $p$ between set of kernels $K$ and $\{k_0\}$, denoted by $I_{\text{diff}}(M, K, \{k_0\}, p)$ is estimated using mask $M_{\text{ref}}$ as given in eq.(6) where $\gamma(p)$ is the correction coefficient of pixel $p$ and $\triangle\delta = \delta(M, p) - \delta(M_{\text{ref}}, p)$.

$$I_{\text{diff}}(M, K, \{k_0\}, p) \approx I_{\text{diff}}(M_{\text{ref}}, K, \{k_0\}, p) + \gamma(p) * \triangle\delta \tag{6}$$

Correction coefficient of each pixel depends on the mask structure around that pixel which is difficult to approximate due to the huge number of possible shapes. However, our experiments show that the correction coefficient is region orientation dependent. A region containing a mask edge parallel to its reference edge is said to be a horizontally oriented region if the contained edge is horizontal, while, it is said to be a vertically oriented region if the contained edge is vertical. Fig.6-b illustrates region types based on orientation.

To find the correction coefficient $\gamma(p)$ without knowing the actual mask structure, linear interpolation is applied as following: Given the original mask $M_{\text{org}}$ (target $T$) with $I_{\text{diff}}(M_{\text{org}}, K, \{k_0\}, p)$ and $\delta(M_{\text{org}}, p)$. One OPC step is applied whose output is a mask $M_{\text{ref}}$. Correction coefficient $\gamma(p)$ for pixel $p$ is calculated as in eq.(7). Note that, if $\delta(M_{\text{org}}, p) = \delta(M_{\text{ref}}, p)$, we set $\gamma(p) = 0$.

$$\gamma(p) = \frac{I_{\text{diff}}(M_{\text{ref}}, K, \{k_0\}, p) - I_{\text{diff}}(M_{\text{org}}, K, \{k_0\}, p)}{\delta(M_{\text{ref}}, p) - \delta(M_{\text{org}}, p)} \tag{7}$$

Once correction coefficients are interpolated, a general OPCed mask $M$ intensity difference map in pixel $p$, is estimated as given in eq.(6) where $\gamma(p)$ is the interpolated correction coefficient for pixel $p$. However, it is time consuming to apply our approach per pixel, therefore, we exploit gridding.

### D. Gridding Layout Region

We divide the pixels region into grids, for each, we consider mask density and average intensity to calculate a grid specific correction coefficient. Thus, the region of pixels is divided into a set of squared grids $W = \{w_1, w_2, ...., w_n\}$ where each grid

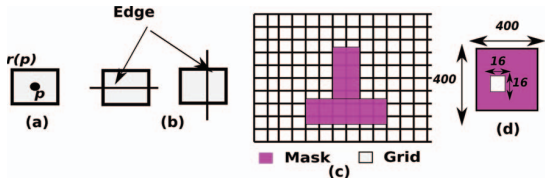



Fig. 6: (a) Region $r(p)$ around pixel $p$. (b) Region Orientation. (c) Region Gridding. (d) A grid trimmed from the mask. All measurements are in nanometer.

area is $Q \times Q$ pixels such that $Q \geq d_{\text{notch}}$. For each grid $w_i$, mask density is the number of mask pixels included in that grid, i.e., $\delta(M, w_i) = |M \cap w_i|$. The average intensity for a grid $w_i$, denoted by $\widehat{I}(M, K, w_i)$, is given in eq.(8). Fig.6-c illustrates gridding process. Grid size is preferred to be chosen such that most of the grids contain one mask edge parallel to its reference edge at maximum.

$$\widehat{I}(M, K, w_i) = \sum_{p \in w_i} I(M, K, p)/|w_i| \tag{8}$$

### E. Kernels Classification

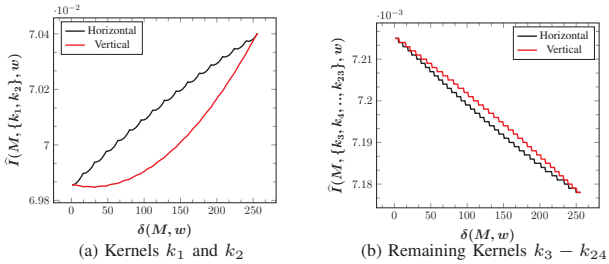
Correction coefficient is grid orientation and kernels dependent. For a grid $w$, the correction coefficient $\gamma(w)$ is interpolated using masks $M_{\text{org}}$ and $M_{\text{ref}}$ as desbribed in eq.(7) but with considering $\widehat{I_{\text{diff}}}(M, K, \{k_0\}, w)$ which is the average intensity differences in $w$. Thus, $\gamma(w)$ is reformulated in eq.(9).

$$\gamma(w) = \frac{\widehat{I_{\text{diff}}}(M_{\text{ref}}, K, \{k_0\}, w) - \widehat{I_{\text{diff}}}(M_{\text{org}}, K, \{k_0\}, w)}{\delta(M_{\text{ref}}, w) - \delta(M_{\text{org}}, w)} \tag{9}$$

Let $K$ be a set of SOCS kernels and $K_s = K - \{k_0\}$ where $k_0$ is the top weight kernel. Roughly saying, if a kernel $k_i \in K_s$ has a large eigenvalue, the correction coefficient is expected to be significantly affected by grid orientation. On the other hand, if $k_i \in K_s$ has a low eigenvalue, correction coefficient can be considered as orientation invariant. In that sense, we classify the kernels in $K_s$ into two sets of kernels: the top weight kernels whose obtained intensity map is expected to be affected by orientation, and the other remaining small kernels. For simplicity, we denote the first set by $K_{\text{ov}}$ (orientation variant) and the latter by $K_{\text{oi}}$ (orientation invariant).

As a case of study, Fig.7 illustrates how average intensity of a grid (shown in Fig.6-d) changes with mask density increase horizontally (fill grid row by row) and vertically (fill grid column by column) for different kernels in lithosim simulator [17] wherein $K = \{k_0, k_1, \ldots, k_{23}\}$ and the top weight kernel is $k_0$. With these results, $K_{\text{ov}} = \{k_1, k_2\}$, and $K_{\text{oi}} = \{k_3, k_4, ..., k_{23}\}$.

With kernels classification, two correction coefficients are utilized: Orientation invariant coefficient, denoted by $\gamma_{\text{oi}}$, which is obtained during preprocessing and it is grid invariant ($\forall w_i \in W, \gamma_{\text{oi}}(w) = \gamma_{\text{oi}}$). The second coefficient is orientation variant coefficient, denoted by $\gamma_{\text{ov}}(w)$, which is grid orientation dependent and linearly interpolated as in eq.(9) with using $K_{\text{ov}} \cup \{k_0\}$ instead of $K$.




(a) Kernels $k_1$ and $k_2$ (b) Remaining Kernels $k_3 - k_{24}$

Fig. 7: Average Intensity VS Mask Density

## VI. Mask Manufacturability Aware Post OPC Modeling

### A. Control Points Defintion

$\forall e' \in E_{\text{r}}$, if $e' \parallel Ref(e')$, we define the control point of edge $e'$, denoted by $c(e')$, as the projection of its center coordinates on the reference edge. This control point controls any further shifting of its edge.

$\forall e' \in E_{\text{nr}}$, its control point $c(e')$ is defined as the center coordinates of the polygon (serif or SRAF) for which it belongs. This control point controls any further enlargement of this polygon. Fig.8-a illustrates control points definition.

### B. EPE Prediction Model

Let $epe_0(T, G(M), c(e'))$ denote EPE value in the control point of edge $e'$ in mask $M$, and let $z(e')$ denotes the orthogonal distance between edge $e'$ and $Ref(e')$ as shown in Fig.8-a. With shifting $e'$ so that this orthogonal distance becomes $z'(e')$ as shown in Fig.8-b, the predicted EPE value in $c(e')$ induced by this shifting, denoted by $epe_{\text{pre}}$, is modeled as in eq.(10) where $I$ is the intensity value in control point, $\triangle z = z' - z$ is shifting distance in $e'$, $\alpha$ is the change in intensity due to edge shifting (found by regression in [4]), $\beta$ is the change in EPE value due to the change in intensity (relation between EPE and intensity is shown in Fig.8-c) which is linearly interpolated by finding intensity map of target $T$ and mask $M$, calculate both EPE and intensity in each edge control point, and apply interpolation as given in eq.(11). $\alpha * \beta(c(e'), T, M)$ is called EPE sensitivity factor.

$$\begin{aligned} &\frac{depe}{dz} = \frac{depe}{dI} \cdot \frac{dI}{dz} = \alpha\beta \Rightarrow depe = \alpha\beta dz \\ &\int_{epe_0}^{epe_{\text{pre}}} depe = \int_{z}^{z'} \alpha\beta dz \\ &epe_{\text{pre}}(c(e'), \triangle z) = epe_0 + \alpha * \beta(c(e'), T, M) * \triangle z \end{aligned} \tag{10}$$

$$\beta(c(e'), T, M) = \frac{epe(T, G(M), c(e')) - epe(T, G(T), c(e'))}{I(M, c(e')) - I(T, c(e'))} \tag{11}$$

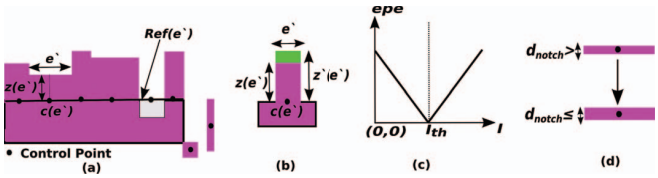



Fig. 8: (a) Control Points. (b) Edge Shifting. (c) Intensity VS EPE in a given point where $epe = \beta|I - I_{\text{th}}|$. (d) SRAF edges Expansion.

### C. Mask Notch Rule Violations Optimization

*1) Core-polygon Edge:* : If edge $e' \in E_{\text{r}}$ violates notch rule and $e' \parallel Ref(e')$, notch violation is resolved by concatenating this edge with its neighboring edge that has the same reference edge and parallel to it. On the other hand, if $e' \perp Ref(e')$, its neighboring edges parallel to $Ref(e')$ are concatenated.

To preserve pattern fidelity during concatenation, EPE prediction model is exploited as follows: Let $e'_i, e'_j$ be two edges to be concatenated to resolve mask notch violation (as in Fig.10-a) with current EPE values $epe_i$ and $epe_j$ in control points $c(e'_i)$ and $c(e'_j)$, respectively and with orthogonal distance $\triangle z_{ij}$ between them. Let $epe_{\text{pre}_i}, epe_{\text{pre}_j}$ be the predicted EPE value calculated as in eq.(10) for each edge control point after assuming that edge to be shifted by $\triangle z_{ij}$ pixels to be concatenated with its neighbor. Concatenation process is applied as following:

- If $epe_{\text{pre}_i} < epe_{\text{pre}_j}$ and $epe_{\text{pre}_i} \leq d_{\text{epe}}$, shift $e'_i$ (Fig.10-b) and set $epe_i = epe_{\text{pre}_i}$.
- if $epe_{\text{pre}_j} < epe_{\text{pre}_i}$ and $epe_{\text{pre}_j} \leq d_{\text{epe}}$, shift $e'_j$ (Fig.10-c) and set $epe_j = epe_{\text{pre}_j}$.
- If $epe_{\text{pre}_i} > d_{\text{epe}}$ and $epe_{\text{pre}_j} > d_{\text{epe}}$, calculate predicted epe due to shifting each edge by $\triangle z_{ij}/2$ towards the other. If no EPE violation is expected, shift both of them (Fig.10-d) and set $epe_i = epe_{\text{pre}_i}$ and $epe_j = epe_{\text{pre}_j}$

- If the predicted EPE causes violation in all the above mentioned cases, no concatenation is performed.
- If concatenation is done, $e'_i$ and $e'_j$ become one edge $e'_{ij}$ whose control point $c(e'_{ij})$ is reassigned and its EPE value is $(epe_i + epe_j)/2$, then proceed to the next neighboring edge to $e'_{ij}$ and repeat above process.

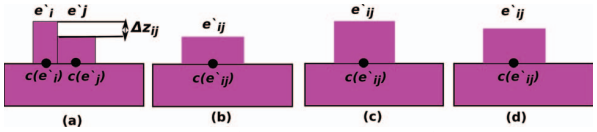



Fig. 9: (a) Original Situation. (b) $e'_i$ is the moving edge. (c) $e'_j$ is moving edge point (d) Both edges are moving to middle location between them.

*2) Serif or SRAF edge:* : If an edge $e' \in E_{\rm nr}$ violates mask notch rule, this edge is simply expanded from both sides so that this violation is resolved as illustrated in Fig.8-d. However, this expansion is subjected to two constraints: (1) No intersection happens between this expanded edge and other patterns. (2) No assisting features printability errors.

Let $e' \in E_{\rm nr}$ be an edge where $c(e')$ is the center of polygon for which $e'$ belongs with center intensity $I(M, c(e'))$. When $e'$ length is increased by $\triangle l(e')$, $I(M, c(e'))$ is expected to increase by $B(\triangle l(e'))$ where $B$ function is obtained by regression. Hence, edge expansion is subjected to the constraint: $I(M, c(e')) + B(\triangle l(e')) < I_{\rm th}$ to avoid printability errors.

### D. Mask Spacing Rule Violations Optimization

Let $(e'_i \in E_M, e'_j \in E_M)$ be a comparison pair of edges where $D(e'_i, e'_j) < d_{\rm min}$, this violation is resolved as follows:

- If $e'_i \in E_{\rm r}$ and $e'_j \in E_{\rm r}$ where $e'_i \parallel e'_j$, each of them is shifted orthogonally by $\triangle z = (d_{\rm min} - D(e'_i, e'_j))/2$ from its current position far from the other (Fig.10-a). This is subjected to the constraint that $epe_{\rm pre}(c(e'_i), \triangle z) < d_{\rm epe}$ and $epe_{\rm pre}(c(e'_j), \triangle z) < d_{\rm epe}$.
- If $e'_i \in E_{\rm r}$ and $e'_j \in E_{\rm r}$ where $e'_i \perp e'_j$, the edge that is parallel to its reference edge will move by $\triangle z = d_{\rm min} - D(e'_i, e'_j)$ from its current position far from the other (Fig.10-b) as long as predicted EPE is beyond $d_{\rm epe}$.
- If $e'_i \in E_{\rm nr}$ and $e'_j \in E_{\rm r}$, the polygon for which $e'_i$ belongs is moved by $\triangle z = d_{\rm min} - D(e'_i, e'_j)$ far from $e'_j$ (Fig.10-c) if no other polygon exists in the new location.
- If $e'_i \in E_{\rm nr}$ and $e'_j \in E_{\rm nr}$, each polygon is moved by $\triangle z = (d_{\rm min} - D(e'_i, e'_j))/2$ far from the other (Fig.10-d) if no other polygon exists in the new locations.
- Once all spacing violations are resolved, wafer image is simulated using adaptive difference map and any SRAF or serif whose movement causes extra EPE violations is restored to its previous state before spacing violations optimization.

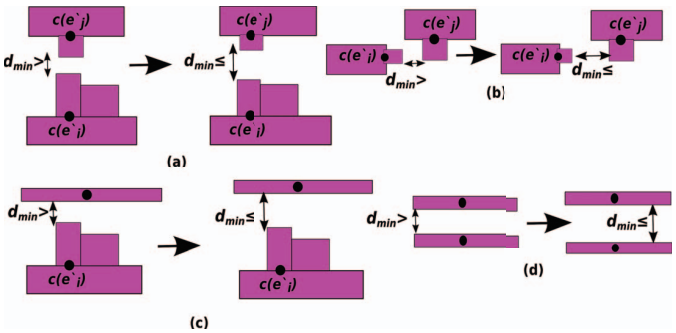



Fig. 10: (a) Moving both edges. (b) Moving only the edge parallel to its reference edge. (c) Moving SRAF. (d) Moving both SRAFs equally.

### E. Mask Data Volume Reduction

Mask data volume is proportional to the number of edges mask contains, therefore, this stage concatenates each pair of neighboring edges that are parallel to their reference as described in concatenation process even if they satisfy mask notch rule. However, this is subjected to the constrain that the predicted EPE value in each edge control point is beyond $d_{\rm epe}$.

## VII. Proposed Algorithm

Algorithm 1 illustrates our proposed algorithm whose input is target pattern $T$, design rules ($d_{\rm notch}$, $d_{\rm min}$, $d_{\rm epe}$), in addition to kernels classification and $\gamma_{\rm oi}$ obtained during preprocessing. Region $R$ is divided into grids wherein grid size is user-defined and must be greater than or equal to $d_{\rm notch}$. Mask $M$ is set to $T$ and its intensity map is simulated using top weight kernel, and $K_{\rm ov}$ to generate $I(M, K_{\rm ov})$. One additional OPC step is applied and new mask density and $I(M, K_{\rm ov})$ are re-calculated to interpolate $\gamma_{\rm ov}(w)$ per grid. Initialization step ends up by using $K_{\rm oi}$ to generate $I(M, K_{\rm oi})$. Note that the total intensity difference map $I_{\rm diff}(M, K, \{k_0\}) = I(M, K_{\rm oi}) + I(M, K_{\rm ov})$.

Algorithm1: Proposed OPC Algorithm

********** Initialization**********
$M \leftarrow T$
$W \leftarrow$ divideRegionIntoGrids($R$)
$I_1(M) \leftarrow$ findIntensityMapUsingTopWeightKernel($M$,$k_0$)
$I(M, K_{\rm ov}) \leftarrow$ findIntensityMapUsingOVKs($M$,$k_{\rm ov}$)
$I(M) \leftarrow I_1(M) + I(M, K_{\rm ov})$
**for** each grid $w_i \in W$ **do**
  $\delta(M, w_i) \leftarrow$ calculateMaskPixelsPerGrid($M, w_i$)
  $\widehat{I}(M, K_{\rm ov}, w_i) \leftarrow$ calculateAverageIntensity($w_i, I(M, K_{\rm ov})$)
**end for**
$M^{[0]} \leftarrow$ applyOPC($M$) //using algorithm described in [4]
$I(M^{[0]}, K_{\rm ov}) \leftarrow$ findIntensityMapUsingOVKs($M$,$k_{\rm ov}$)
**for** each grid $w_i \in W$ **do**
  $\delta(M^{[0]}, w_i) \leftarrow$ calculateMaskPixelsPerGrid($M^{[0]}, w_i$)
  $\widehat{I}(M^{[0]}, K_{\rm ov}, w_i) \leftarrow$ calculateAverageIntensity($w_i, I(M, K_{\rm ov})$)
  $\gamma_{\rm ov}(w_i) \leftarrow$ interpolate($\delta(M, w_i), \delta(M^{[0]}, w_i), \widehat{I}(M, K_{\rm ov}, w_i), \widehat{I}(M^{[0]}, K_{\rm ov}, w_i)$)
**end for**
$I(M, K_{\rm oi}) \leftarrow$ findIntensityMapUsingOIKs($M$,$K_{\rm oi}$)
$I(M) \leftarrow I_1(M) + I(M, K_{\rm oi}) + I(M, K_{\rm ov})$
$i \leftarrow 1$
********** EPE Optimization**********
**while** epe stop conditions are not satisfied **do**
  $M^{[i]} \leftarrow$ applyOPC($M^{[i-1]}$) //shifting, hammering, SRAF insertion
  $I(M^{[i]}) \leftarrow$ findAdaptiveIntensityMap($M^{[i]}, M^{[i-1]}, W, \gamma_{\rm oi}, \gamma_{\rm ov}, I(M^{[i]})$)
  $G_n(M^{[i]}) \leftarrow$ applyResistModel($I(M^{[i]}), I_{\rm th}$)
  $epe \leftarrow$ calculateEPE($G_n(M^{[i]}), T$)
  $i \leftarrow i + 1$
**end while**
$M \leftarrow$ chooseMaskWithLeastEPE($M[i]$)
********** Post OPC**********
$I(M) \leftarrow$ findAdaptiveIntensityMap($M, M, W, \gamma_{\rm oi}, \gamma_{\rm ov}, I(M)$)
$Sen \leftarrow$ calculateControlPointsAndEPESensitivity($M, T$)
$M^* \leftarrow$ minimizeMaskNotchV($M, I(M), Sen, d_{\rm notch}$)
$I(M^*) \leftarrow$ findAdaptiveIntensityMap($M^*, M, W, \gamma_{\rm oi}, \gamma_{\rm ov}, I(M)$)
$Sen \leftarrow$ calculateControlPointsAndEPESensitivity($M^*, T$)
$M^{**} \leftarrow$ minimizeSpacingV($M^*, I(M^*), d_{\rm min}$)
$I(M^{**}) \leftarrow$ findAdaptiveIntensityMap($M^{**}, M, W, \gamma_{\rm oi}, \gamma_{\rm ov}, I(M)$)
$M_o \leftarrow$ reduceDataVolume($M^{**}, Sen$)

Thereafter, EPE optimization starts in which OPC steps are executed followed by calculating the mask density for each grid to be used along with the correction coefficients to correct the intensity value for each pixel after simulation using only top weight kernel as shown in Algorithm 2. Note that, the estimated change in average intensity is applied for each pixel inside the grid. CTR model is then used to obtain wafer image $G(M)$ for EPE calculation. Once EPE violations number reaches zero or we exceed the maximum number of iterations, least cost mask $M$ is outputted.

Finally, OPCed mask $M$ undergoes post processing as following: For each edge in mask $M$, control point is determined for which EPE sensitivity factor is calculated. Thereafter, $M$ undergoes mask notch violations optimization stage by concatenating neighboring segments and SRAFs/serif enlargement. The output is mask $M^*$ whose intensity map is simulated using Algorithm 2 to recalculate EPE sensitivity factor for

Table 1: Comparison between our algorithm and work published in [4]

| Benchmark | Algorithm Published in [4] | | | | | Our Algorithm | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | #NotchV | #SpaceV | V(M) | #EPEV | Time | #NotchV | #SpaceV | V(M) | #EPEV | Time |
| B1 | 112 | 14 | 6988 | 8 | 86 | 25 | 10 | 5745 | 5 | 104 |
| B2 | 80 | 16 | 5618 | 3 | 78 | 19 | 8 | 4640 | 2 | 110 |
| B3 | 126 | 30 | 8909 | 42 | 87 | 23 | 14 | 7481 | 33 | 110 |
| B4 | 47 | 14 | 3140 | 0 | 81 | 4 | 5 | 2300 | 0 | 105 |
| B5 | 105 | 28 | 6508 | 6 | 87 | 12 | 16 | 5054 | 2 | 108 |
| B6 | 108 | 22 | 6290 | 3 | 92 | 25 | 11 | 5267 | 2 | 112 |
| B7 | 62 | 13 | 4258 | 0 | 74 | 24 | 6 | 3085 | 0 | 104 |
| B8 | 37 | 6 | 3270 | 0 | 80 | 6 | 2 | 2568 | 0 | 96 |
| B9 | 122 | 26 | 7401 | 6 | 80 | 40 | 14 | 5729 | 0 | 106 |
| B10 | 18 | 0 | 2242 | 0 | 78 | 0 | 0 | 1868 | 0 | 95 |
| **Average** | **81.7** | **16.9** | **5462.4** | **6.8** | **82.3** | **17.8** | **8.6** | **4373.7** | **4.4** | **105** |
| **Ratio** | **4.6** | **2.0** | **1.2** | **1.5** | **0.8** | **1.0** | **1.0** | **1.0** | **1.0** | **1.0** |

Time unit:Sec, $V(M)$:Data Volume in bytes, Fragment length= 20 nm for both experiments.

Algorithm 2:findAdaptiveIntensityMap($M$, $M_{\mathrm{org}}$, $W$, $\gamma_{\mathrm{oi}}$, $\gamma_{\mathrm{ov}}$, $I(M)$)

$I_1(M) \leftarrow$ findIntensityMapUsingTopWeightKernel($M$)
**for** each grid $w_i \in W$ **do**
  $\delta(M, w_i) \leftarrow$ calculateMaskPixelsPerGrid($M$, $w_i$)
  $\triangle\delta \leftarrow \delta(M, w_i) - \delta(M_{\mathrm{org}}, w_i)$
  **for** each pixel $p \in W_i$ **do**
    $I(M, K_{\mathrm{oi}}, p) \leftarrow I(M_{\mathrm{org}}, K_{\mathrm{oi}}, p) + \triangle\delta * \gamma_{\mathrm{oi}}$
    $I(M, K_{\mathrm{ov}}, p) \leftarrow I(M_{\mathrm{org}}, K_{\mathrm{ov}}, p) + \triangle\delta * \gamma_{\mathrm{ov}}(w_i)$
    $I_1^*(M, p) \leftarrow I_1(M, p) + I(M, K_{\mathrm{oi}}, p) + I(M, K_{\mathrm{ov}}, p)$
  **end for**
**end for**
$I(M) \leftarrow I_1^*(M)$

each new edge control point. $M^*$ undergoes then mask spacing violations optimization in which edges are shifted and SRAFs are moved slightly if necessary. The output of this module is mask $M^{**}$ which is forwarded to data volume optimization in which concatenation process is applied if possible following EPE prediction model. The final output is mask $M_o$.

## VIII. Experimental Results

We implemented our algorithm on top of lithosim simulator from ICCAD 2013 CAD contest [17]. In this simulator, a target pattern $T$ is defined in $1024 \times 1024$ pixels region where each pixel represents 1 nm $\times$ 1 nm, $I_{\mathrm{th}} = 0.225$ and, total number of kernels $|K| = 24$. During OPC, shifting, hammering and SRAFs insertion were implemented using algorithm published in [4] with 20 nm segment (fragment) length and 10 iterations number. Grid size was chosen 8 nm $\times$ 8 nm, $d_{\mathrm{epe}}$=15 nm, $d_{\mathrm{notch}}$=5 nm, and $d_{\mathrm{min}}$=10 nm.

OPC algorithm published in [4] and our proposed algorithm were executed on public benchmarks released by IBM for ICCAD contest. Experimental results are shown in Table 1 which includes, notch violations number, spacing violations number, mask data volume, EPE violations number (calculated using EPE checker tool [17]), and computational time.

As shown in Table 1, notch violations and spacing violations are reduced by around 78% and 50%, respectively and mask data volume is reduced by around 20%. Additionally, image quality is preserved with 35% reduction in EPE violations in average because of adaptive intensity difference map usage and EPE prediction model. Computational time has increased by 20% in average. Fig.11 and Fig.12 show portions of the masks obtained using our algorithm and algorithms published in [4][6][15] on the same benchmarks which show that our masks are less complex in general with almost same EPE violations number as others. Moreover, our post processing stage is generic and reusable to improve manufacturability of an OPCed mask generated by any other OPC algorithm.

## IX. Conclusions and Future Work

In this paper, we proposed a fast algorithm to iteratively improve wafer image estimation during OPC for better pattern fidelity. Thereafter, we proposed a post processing stage to improve mask manufacturability with preserving acceptable pattern fidelity . Our experimental results show the effectiveness of our algorithm. In future work, we will consider process window preservation as well and hotspots fixing.

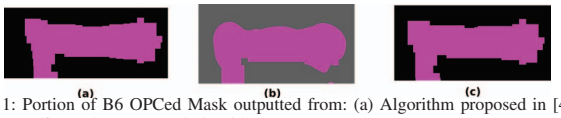


Fig. 11: Portion of B6 OPCed Mask outputted from: (a) Algorithm proposed in [4]. (b) MOSAIC [6] (c) Our proposed algorithm.

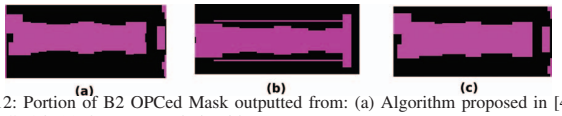


Fig. 12: Portion of B2 OPCed Mask outputted from: (a) Algorithm proposed in [4]. (b) PVOPC [15] (c) Our proposed algorithm.

## References


[1] Xu Ma and Gonzalu Arce, *"Computational Lithography"*, Wiley Publisher, 2010.

[2] R.Fabian Pease and Stephen Chou, *"Lithography and Other Patterning Techniques for Future Electronics"*, proc.IEEE, pp.248-270, 2008.

[3] Shayak Banerjee, Zhou Li and San Nassif, *"ICCAD-2013 CAD Contest in Mask Optimization abd Benchmark Suite"*, proc.ICCAD, pp.271-274, 2013.

[4] Ahmed Awad, Atsushi Takahashi, Satoshi Tanaka, and Chikaaki Kodama, *"A Fast Process Variation and Pattern Fidelity Aware Mask Optimization Algorithm"*, proc.ICCAD, pp.238-245, 2014.

[5] L.Capodieci, P.Gupta, A.Kahng,D.Sylvester, and J.Yang , *"Toward a Methodology for Manufacturability-Driven Design Rule Exploration"*, proc.DAC, pp.311-316, 2004.

[6] Jhih-Rong Gao, Xiaoqing Xu, Bei Yu, and David Pan, *"MOSAIC: Mask Optimization Solution With Process Window Aware Inverse Correction"*, proc.DAC, pp.1-6, 2006.

[7] Satomi Shiori and Hiroyoshi Tanabe, "Fast Optical Proximity Correction: Analytical Method", Proc. SPIE 2440,pp. 261-269, 1995.

[8] Gregg Gallatin, Kafai Lai, Maharaj Mukhejree, Alan Rosenbluth, *"Printability Verification By Progressive Modeling Accuracy"*, US Patent 7512927, 2009.

[9] Linyong Pang, Yong Liu, and Dan Abrams, *"Inverse Lithography Technology (ILT), What is the Impact to Photomask Industry?"*, Proc. SPIE 6283, Photomask and Next-Generation Lithography Mask Technology, 2006.

[10] Nick Cobb and Yuri Granik, and David Pan, *"Model-based OPC using the MEEF matrix"*, proc.Annual BACUS Symposium on Photomask Technology, pp.1281-1292, 2002.

[11] Peng Yu and David Pan, *"A Novel Intensity Based Optical Proximity Correction Algorithm with Speedup in Lithography Simulation"*, proc.ICCAD, pp.854-858, 2007.

[12] Puneet Gupta, Andrew Kahng, Swamy Muddu, Sam Nakagawa, and Chul-Hong Park *"Modeling OPC Complexity for Design for Manufacturability"*, Proc. SPIE, vol.5992, 2005.

[13] Tadao Yasuzato, *"Mask Data Creation Method"*, US Patent 7691543B2, 2010.

[14] Yeonah Shim, Jaeyoung Choi, Jeahee Kim, Bo Su, Ping Zhang, and Keun Kim, *"Improvement on OPC Completeness through pre-OPC Hot Spot Detection and Fix"*, Proc. SPIE, vol.6925, 2008.

[15] Yu-Hsuan Su, Yu-Chen Huang, Liang-Chun Tsai, Yao-Wen Chang, and Shayak Banerjee, *"Fast Lithographic Mask Optimization Considering Process Variation"*, proc.ICCAD, pp.230-237, 2014.

[16] Nick Cobb, *"Sum of Coherent Systems Decomposition by SVD"*, 1995.

[17] http://cad_contest.cs.nctu.edu.tw/CAD-contest-at-ICCAD2013/problem_c/