# Mask Manufacturability Aware Post OPC Algorithm For Optical Lithography

Ahmed Awad and Atsushi Takahashi
*Department of Communications and Computer Engineering, Tokyo Institute of Technology, Tokyo, Japan
Email: {awad.ahmed, atsushi}@eda.ce.titech.ac.jp

***Abstract*—As technology nodes continue scaling down into sub-20nm features, aggressive Optical Proximity Correction (OPC) is the main stream to preserve feature fidelity in silicon wafer for the foreseeable future of optical micro-lithography. Although high level of aggressiveness during OPC is required for better circuit performance in terms of timing and power, it results in complex mask patterns which is directly proportional to mask manufacturability costs, such as, mask writing time and mask data volume. Furthermore, unfriendly litho-patterns might lead to hot spots after OPC. To consider mask manufacturability, several algorithms have been proposed in the field of design aware OPC. In some algorithms, intensive timing and power study on the target circuit is applied prior to OPC to recognize critical regions on which strict OPC is applied while relaxed OPC is applied on the other regions. Other algorithms push the trade-off between mask manufacturability and circuit performance in favor of mask manufacturability, sacrificing parametric yield, through adding a set of restricted design rules to be taken into consideration during OPC response. However, for advanced small-sized dense layouts, design aware OPC algorithms should be executed carefully to ensure pattern fidelity without causing circuit malfunction or significant performance degradation. Therefore, in this paper, we propose a new post processing algorithm, whose objective is to minimize OPCed mask manufacturability costs with pattern fidelity and process window preservation. Our algorithm considers features spacing, mask notch, assisting features dimensions and jogs as design rules whose violations are penalized in a cost function that the algorithm aims to minimize subjected to the constraints that preserve circuit performance within its allowable tolerance.**

## I. Introduction

With the advances in Very Large Scale Integration(VLSI) technology, advanced nanometer nodes scaling down is being pushed into sub-20nm regime. Consequently, optical lithography becomes more and more challenging during Integrated Circuits (ICs) fabrication process. In optical lithography, an IC is patterned layer by layer through projecting the image of a pixelated template, called a mask (on which circuit layout is carved as a set of polygons), onto a light sensitive polymer (photoresist) coating the silicon wafer. If light intensity reaches a certain threshold, the resist is chemically exposed and etched under development process to transfer the target pattern onto silicon wafer.[1].

To keep pace with the continuous shrinkage of pattern features dimensions, the wavelength of optical lithographic system has been steadily reduced till it reaches its practical limit at 193nm immersion lithography. As a result, mask image quality for small sized features suffers from quality degradation, such as shortened lines, rounded corners, holes, and bridging errors. The impact of those distortions on circuit functionality and performance could be significant. Since Next Generation Lithography (NGL), such as electron beam and Extreme Ultra Violet Light (EUVL) is still not in the track, the industry relies heavily on Resolution Enhancement Techniques (RETs) to improve image quality against distortions [2].

Optical Proximity Correction (OPC) is one of the dominant RETs for the foreseeable future. In OPC, mask image onto the silicon wafer is simulated using an optical lithographic model. This simulated image is compared with the target pattern. If it is deviated from the target pattern (golden image), mask pattern is adjusted and its image is re-simulated. This iterative process continues till mask image deviation from the golden image becomes within the allowable tolerance.[2].

Image quality is typically evaluated by calculating the geometrical distance between any point on the target layout and its corresponding point on the mask image. This distance is called Edge Placement Error (EPE) which is usually statically measured on various points on the boundary of the layout. If EPE distance is greater than allowable tolerance, it is considered as violation. The more violations, the more impact on circuit performance in terms of timing and power. Fig.1 illustrates EPE examples and some kinds of image distortions in optical lithography.

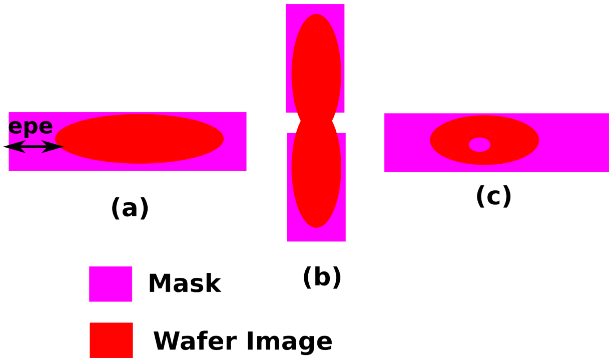

Fig. 1: EPE measurement and distortion examples: (a) Shortened lines. (b) Bridging Error. (c) Hole.

Several algorithms have been proposed in the field of OPC to minimize the number of EPE violations. However, to preserve circuit performance for advanced technology nodes, high level of aggressiveness is required during OPC to minimize the number of EPE violations. This includes, shorter segments lengths, more variations in segments shifting distances, more SRAFs jogs and other patterns. Consequently, complex mask pattern will be outputted as optimal mask solutions which abruptly increases mask manufacturability cost in terms of mask writing time and data volume. To consider mask manufacturablity, Restricted Design Rules (RDRs) have been introduced to define minimum manufacturable features dimensions in addition to the minimum spacing between patterns in the layout [3]. However, considering RDRs during OPC would significantly increase OPC computational time for small sized dense layouts in advanced technology nodes due to the expected low stability and low convergence in finding optimal mask solution in terms of EPE and RDR violations. Furthermore, defining only minimum manufacutrable dimensions in not sufficient. For example, two features length can be manufactured (satisfy mask notch) while the orthogonal distance between them is too small to be manufactured as shown in Fig.2.

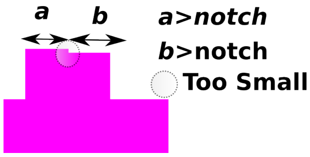

Fig. 2: EPE measurement and distortion examples: (a) Shortened lines. (b) Bridging Error. (c) Hole.

In this paper, we propose a new post OPC processing algorithm which aims to reduce Mask Manufacturability Cost (MMC) for an OPCed mask in terms of Restriced Design Rules (RDRs) violations and mask data volume within a short computational time and without significant increase in EPE violations to preserve circuit functionality and performance. Our contributions are summarized as following:

- A post OPC professing algorithm is proposed to reduce mask manufacturability cost of an OPCed mask within a short computational time and without significant EPE violations increase. This is achieved through SRAFs alignment, SRAFs movement, segments shifting alignment, and segments concatenation following a new EPE prediction model.
- Our proposed algorithm reduces the cost defined in terms of mask manufacturablity, EPE, and computational time effectively on the public benchmarks released by IBM.

The rest of this paper is organized as following: Previous work is presented in Section II. Lithographic terminology and evaluation parameters are proposed in Section III. Our post OPC algorithm overview is proposed in Section IV and system modeling is proposed in Section V followed by detailed algorithm in Section VI. Section VII shows our experimental results and Section VIII concludes this paper.

## II. Previous Work

Several algorithms have been proposed for model based OPC that effectively minimize EPE violations such as the huge body of work in inverse lithography [4][5], intensity based OPC [6], and Mask Error Enhancement Factor (MEEF) matrix usage [7]. However, with applying those algorithms on advanced technology nodes, complex mask patterns are expected to be outputted as optimal mask solutions, resulting in large mask manufacturability cost.

To consider mask manufacturability, design aware OPC has been proposed in which intensive timing/power analysis is applied on the circuit to define the most critical regions on which aggressive OPC is applied while more relaxed OPC is applied on the other regions [8]. However, such an approach is circuit dependent and it will be time consuming to do timing/power analysis for each target circuitry. Another algorithm was proposed in [9] in which pre OPC step is applied to detect and eliminate hot spots regions for better manufacturability during the OPC. However, mask manufacturability is expected to increase during OPC for advanced technology nodes. Applying Restricted Design Rules (RDRs) that define minimum features dimensions (mask notch) and minimum spacing during OPC recipe was proposed in [3][10]. However, algorithm convergence is expected to be low in addition to large computational time to find optimal mask solution in terms of EPE and RDR violations. SRAFs manufacturability was also considered in [11] at the cost of more EPE violations.

In this paper, we propose a post OPC processing algorithm to improve mask manufacturability for an OPCed mask generated during a fast OPC algorithm. Our algorithm aims to minimize mask manufacturability cost within a short computational time without significant EPE violations increase to preserve circuit performance.

## III. Preliminaries

### A. Problem Formulation

Given a target pattern layout and an OPCed mask for that pattern defined in a region of pixels with minimal number of EPE violations, the main objective is to minimize Mask Manufacturability Cost (MMC) of that OPCed mask in terms of Restricted Design Rules (RDRs) violations and mask data volume without significant increase in EPE violations number and within a short computational time.

### B. Lithographic Terminology

Let $R$ be a region of $N \times N$ pixels in which a target pattern $T$ and mask $M$ are defined where $T \subset R$ and $M \subset R$. A target pattern $T$ consists of a number of polygons where a polygon $S \in T$. If a pixel $p \in S \in T$, we simply denote it $p \in T$. The boundary of each polygon in $T$ is fragmented into movable segments whose centers are defined as tap points. The set of tap points $A = \{t_0, t_1, ..., t_{n-1}\}$ is used to guide segment shifting during OPC in addition to mask evaluation as will be explained later. Fig.3 illustrates fragmentation process.

In this paper, we extend the OPC algorithm published in [12] in which an OPCed mask consists of three components: core-polygons, hammers and Sub Resolution Assisting Features (SRAFs) as illustrated in Fig.4. Core-polygons are modified versions of the polygons in target pattern $T$ constructed after segments movement to improve their printablity. A hammer (serif) is a square inserted closed to a corner to make it tolerant against rounded corners. SRAFs are long unprintable bars inserted parallel to target pattern polygons edges to improve process window.

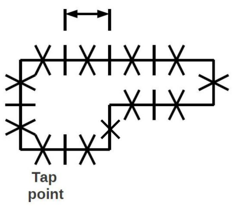

Fig. 3: Layout Fragmentation[12].

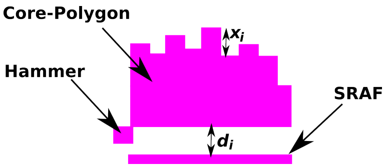

Fig. 4: Mask Components.

### C. Lithographic Model

To simulate mask image onto silicon wafer, the optical and projection systems are modeled following Hopkins model of light intensity in which a 4-D integral is used to approximate light intensity in a pixel $p$ through mask $M$. However, this model is bilinearly transformed into 2-D convolution in the spatial domain after decomposing the optical system into a set of coherent kernels working as low pass filters. The mask $M$ is transformed into spatial frequency spectrum using Fast Fourier Transform (FFT) and then undergoes filteration process with each kernel. Let $K = \{k_0, k_1, ..., k_{|K|-1}\}$ be a set of kernels, a kernel $k_i \in K$ has an eigen function $\phi_{k_i}$ and eigenvalue $\sigma_{k_i}$ which represents its weight or contribution for light intensity. By this way, intensity map of mask $M$, denoted by $I(M)$ is obtained as in eq.(1). For a pixel $p$, intensity value is denoted by $I(M,p)$. This model is called Sum of Coherent Systems (SOCS) which is commonly used in OPC algorithms [12]. Once intensity map is obtained, Constant Threshold Resist (CTR) model is applied to extract mask image onto silicon wafer. For a mask $M$, mask image $G(M)$ is the set of exposed pixels in the intensity map, i.e $G(M) = \{p \in R | I(M,p) \geq I_{\text{th}}\}$ where $I_{\text{th}}$ is the target or threshold intensity.

$$I(M) = \sum_{i=1}^{|K|} \sigma_{k_i} |(\phi_{k_i} \otimes M)|^2 \tag{1}$$

### D. EPE Formulation

For a tap point $t$, $epe(T, G(M), t)$ is the geometrical distance between point $t$ in the target pattern $T$ and its corresponding point in mask image $G(M)$. Ideally, a mask $M$ has no EPE if and only of $\forall p \in T, epe(T, G(M), p) = 0$. However, with the current available optical lithographic system, satisfying such a condition is infeasible for advanced technology nodes, therefore, EPE is relaxed to be allowable within a certain tolerance as long as no errors causing circuit malfunction occur (such as holes or bridging errors). Let $epe_{\max}$ be the maximum allowable distance for EPE. In that sense, the number of EPE violations for a mask $M$ is given in eq.(2) where $A$ is the set of tap points defined in target pattern $T$.

$$\begin{gathered} isEPE(T,G,t) = \begin{cases} 1; & epe(T,G,t) > epe_{\max} \\ 0; & \text{Otherwise} \end{cases} \\ \#EPEV(M) = \sum_{t \in A} isEPE(T, G(M), t) \end{gathered} \tag{2}$$

### E. Restricted Design Rules (RDRs) Formulation

Mask manufacturability is considered in terms of Restricted Design Rules (RDRs) violations and mask data volume. However, RDRs definition in the literature, which defines only minimum feature dimensions and spacing, is not sufficient for good mask manufacturability consideration for advanced technology nodes. Therefore, we strictly define RDRs violations as following: Let $X = \{x_0, x_1, ...., x_n\}$ be a set of features in mask $M$ (such as $x_i$ in Fig.4), and let $D = \{d_0, d_1, ..., d_m\}$ be set of distances between different patterns (such as $d_i$ in Fig.4), feature $x_i$ violates RDRs if its length, denoted by $|x_i|$ is not one of the multiples of the base manufacturable distance $x_{\text{th}}$ defined by design rules. Similarly, a distance $d_i$ between two patterns is said to be RDR violation if $d_i < d_{\text{th}}$ where $d_{\text{th}}$ is the minimal allowable spacing distance between patterns defined by design rules. In that sense, Restricted Design Rules Violation number for mask $M$, denoted by $\#RDRV(M)$, is given in eq.(4).

$$f(x) = \begin{cases} 0; & |x| \bmod x_{\text{th}} = 0 \\ 1; & \text{Otherwise} \end{cases} \quad ; \quad q(d) = \begin{cases} 1; & d < d_{\text{th}} \\ 0; & \text{Otherwise} \end{cases} \tag{3}$$

$$\#RDRV(M) = \sum_{x \in X} f(x) + \sum_{d \in D} q(d) \tag{4}$$

## IV. Algorithm Overview

Fig.5 illustrates the flowchart of our post OPC algorithm whose input is a target patter $T$ and OPCed mask $M$ and the output is a modified version of mask $M$. First of all, layout region is divided into grids whose borders are used as alignment baselines in the proceeding steps. Then, SRAFs alignment and movement is applied to make their sizing and spacing within the desired design rules outputting mask $M^*$. The third step is segments shifting alignment to eliminate small shifting distances and small orthogonal distances between neighboring segments whose output is mask $M^{**}$. Finally, segments concatenation step is applied to reduce mask file size and improve layout regularity. This algorithm will be described in details in Section VI.

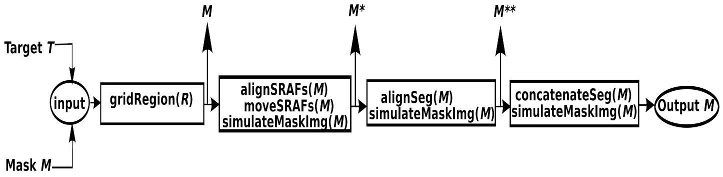

Fig. 5: Flowchart of Post-OPC Algorithm

## V. System Modeling

### A. Layout Gridding and Base Dimensions Choice

The purpose of gridding is to allow features to be aligned on grids borders where each grid has the dimensions $x_{\text{th}} \times x_{\text{th}}$ pixels where $x_{\text{th}}$ is the base manufacturable distance. By this way, region $R$ is divided into $N^2/x_{\text{th}}^2$ grids and the distance between each feature $x_i$ and its closest grid border, denoted by $\triangle x_i$, is calculated as shown in Fig.6 and eq.(5). The minimum spacing distance between different patterns depends on the user defined design rules, however, it is subjected to the constraint $d_{\text{th}} \geq x_{\text{th}}$

$$\triangle x_i = x_{\text{th}} - |x_i| \bmod x_{\text{th}} \tag{5}$$

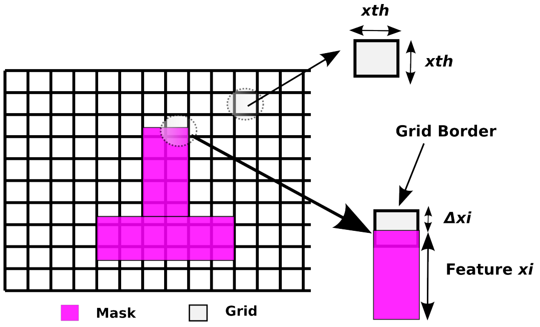

Fig. 6: Grids and Features.

### B. Feature Alignment With Grid Borders

To improve mask manufacturablity, a feature $x_i$ can be aligned onto grid borders. However, EPE violations number is expected to abruptly increase if add hoc alignment is applied. Therefore, this alignment should be subjected to some constraints to preserve circuit performance. The type of constraint is feature dependent. For example, SRAF alignment should ensure avoiding SRAFs printablity or side lobe errors. On the other hand, core-polygon segments shifting should ensure that the EPE value in the tap points of those segments is still within the allowable tolerance.

*1) SRAF Alignment:* Let $Sr = \{Sr_0, Sr_1, ..., Sr_{m-1}\}$ be set of SRAFs in mask $M$. An SRAF becomes manufacturable by simply expanding all its edges to be aligned within the grids borders as shown in Fig.7. However, with such alignment, the intensity induced by SRAF is expected to increase resulting in undesirable printablity or side lobe errors specially when SRAFs are inserted between polygons in dense layouts. Therefore, this alignment should be subjected to a constraint that ensures this unprintablity. This is achieved by considering the maximum current intensity of the SRAF (before alignment). If this intensity is less than the a certain intensity value (usually $I_{\text{th}} - B$ where $B$ is upper bound value), then the SRAF is allowed to be aligned, otherwise, it will keep its current configuration. eq.(7) illustrates SRAFs alignment formulation where $I_{\max}(Sr_i)$ is the maximum intensity in SRAF pixels and $B$ is some user defined upper bound value to prevent SRAF printability. Note that $I(M,p)$ in eq.(6) is obtained by simulating the OPCed mask before post processing. By this way, the output of SRAFs alignment step is a set of aligned and nonaligned SRAFs included in mask $M^*$ as shown in Fig.(5).

$$I_{\max}(Sr_i) = \max(\{I(M,p) : \forall p \in Sr_i\}) \tag{6}$$

$$\forall Sr_i \in Sr,\ \ \text{align}(Sr_i) \quad \textbf{Subject to} \ \ I_{\max}(Sr_i) \le I_{\text{th}} - B \tag{7}$$

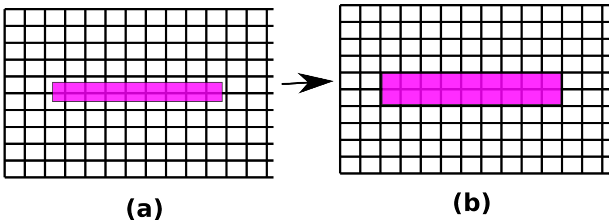

Fig. 7: (a) SRAF before re-sizing. (b) SRAF after re-sizing

*2) Segment Alignment:* Let $s = \{s_0, s_1, ...s_{n-1}\}$ be set of movable segments defined in the layout where a segment $s_i$ has tap point $t_i$ in its center in the target pattern $T$. Each segment has its current position $x(s_i)$ (with considering the origin at the segment position in the target $T$) and $\triangle x(s_i)$ represents the distance between the current position of segment $s_i$ and its closest grid as shown in Fig.8. The manufacturability is improved when segments are aligned on the borders of the grids. However, EPE in the tap point of each segment is expected to be affected. Therefore, shifting a segment to its closest grid border is subjected to the constraint that the expected EPE in its tap point should remain within the allowable EPE distance. By this way, segment alignment problem is formulated as in eq.(8) where $epe_{\text{ex}}(T, G, t_i, \triangle x(s_i))$ is the expected EPE in tap point $t_i$ after shifting its segment $s_i$ by $\triangle x(s_i)$ distance and $x'(s_i)$ denotes the new segment shifting after alignment. The expected EPE value is calculated based on the model described in the following subsection in addition to the simulated image of mask $M^*$ which is the output of SRAFs alignment and movement process (see Fig.(5)

$$\begin{gathered} \forall s_i \in s,\ \ x'(s_i) = x(s_i) + \triangle x(s_i) \\ \textbf{Subject to} \\ epe_{\text{ex}}(T, G, t_i, \triangle x(s_i)) \le epe_{\max} \end{gathered} \tag{8}$$

The output of segments alignment step is new positions of the segments to be aligned on their closed grids borders. However, if the expected EPE value for the tap point of any segment exceeds the allowable EPE distance, it will not move. Therefore, some segments are expected to stay in their positions in the OPCed mask after segments alignment to avoid significant increase in EPE violations. $M^{**}$ is the output of segments alignment step.

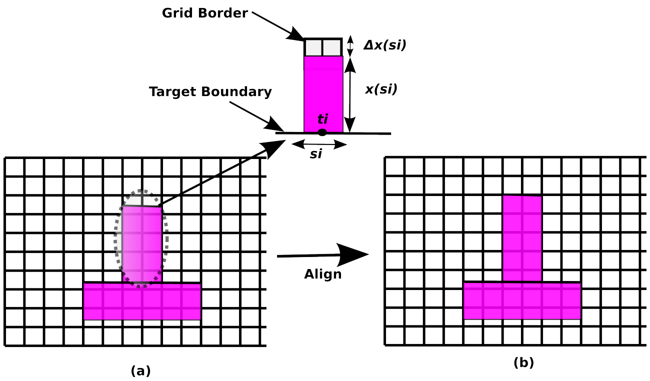

Fig. 8: (a) Before Segment Alignment . (b) After Segment Alignment.

### C. EPE Prediction Model

One of the key points is how to predict the EPE in a tap point as a response to its segment position change. Generally, applying alignment on all segments and then simulate mask image followed by restoring those who caused EPE violations to their previous position is unfair. The reason is that, a segment would affect its neighbor, and all segments alignment is expected to significantly cause image quality degradation without identifying the exact segments causing huge violations. Therefore, we propose an intensity based prediction model to expect EPE value after alignment for each segment individually before proceeding to its neighbor. Fig.9 illustrates the relation between EPE and intensity in a given tap point where the rate of this change can be approximated by $\alpha(t)$ which is segment dependent.

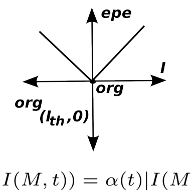

Fig. 9: $epe(I(M,t)) = \alpha(t)|I(M,t) - I_{\text{th}}|$

eq.(9) shows our differential model to calculate the expected EPE value for tap point $t$ after shifting its segment by $\triangle x(s)$ distance to be aligned on its closest grid borders. $epe_{\text{ex}}$ is the predicted value while $epe_0$ is the current epe value in the tap point before alignment ($epe_0 = epe(T, G(M^*), t)$. $\beta$ is the rate of change in the intensity in tap point $t$ after shifting its segment $s$ by $x(s)$ pixels from its original location in the target. This value was found by regression as published in [12]. Regarding $\alpha(t)$ which is the rate of change in the EPE of a tap point with the change in its intensity, it can be linearly interpolated as following: Given tap point $t$ in target pattern $T$ with intensity $I(T,t)$ and EPE value $epe(T,G(T),t)$ after simulating target image. Similarly, the OPCed mask $M$ image is simulated and both new intensity $I(M,t)$ and EPE $epe(T,G(M),t)$ are calculated. The constant $\alpha(t)$ is calculated for each tap point (segment) as in eq.(10). Consequently, the predicted EPE value in a tap point $t$ after shifting its segment $s$ by $\triangle x(s)$ distance from its current position can be approximated as in eq.(11)

$$\begin{aligned}
&\frac{depe}{dx} = \frac{depe}{dI}.\frac{dI}{dx} = \alpha\beta \\
&depe = \alpha\beta dx \\
&\int_{epe_0}^{epe_{\mathrm{ex}}} depe = \int_{x}^{x+\triangle x} \alpha\beta \\
&epe_{\mathrm{ex}}(t,\triangle x) = epe_0 + \alpha(t)\beta\triangle x
\end{aligned} \tag{9}$$

$$\alpha(t) = \frac{epe(T,G(M),t) - epe(T,G(T),t)}{I(M,t) - I(T,t)} \tag{10}$$

$$epe_{\mathrm{ex}}(t,\triangle x) \approx epe_0 + \frac{epe(T,G(M),t) - epe(T,G(T),t)}{I(M,t) - I(T,t)}\beta\triangle x \tag{11}$$

### *D. SRAFs Movement*

If the spacing distance between SRAF and other patterns is less than $d_{\mathrm{th}}$, it will be shifted in the opposite direction till it satisfies the minimum spacing distance design rule. However, in some OPC algorithms, SRAFs are placed closed to core-polygons to increase intensity value. Therefore, if a moved SRAF causes EPE violations in its neighboring segments tap points, it will be restored to its previous state. Such an impact is detected by simulating mask $M^*$ image after moving all SRAFs and re-calculating the EPE in each tap point followed by restoring those SRAFs that caused violations. Fig.10 illustrates an SRAF movement in which epe values in tap points $t_0$ and $t_1$ are assumed to remain beyond the allowable EPE value.

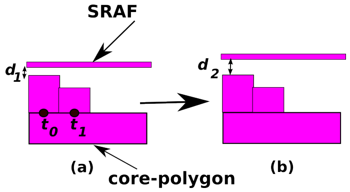

Fig. 10: (a) SRAF before shifting where $d_1 < d_{\mathrm{th}}$. (b) SRAF after shifting where $d_2 \geq d_{\mathrm{th}}$

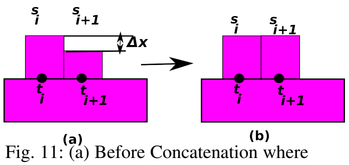

Fig. 11: (a) Before Concatenation where $epe(t_i) > epe(t_{i+1})$. (b) After Concatenation.

### *E. Segments Concatenation*

The purpose of this step is to reduce the number of segments and increase pattern regularity for less data volume. For each pair of segments in mask $M^{**}$ (which is the output of segments alignment as illustrated in Fig.5), $s_i$ and $s_{i+1}$, the segment whose intensity has less epe is the candidate to move. This segment is shifted to be in the same position of its neighbor. The predicted epe is calculated as in the previously described prediction model in eq.(11). If no violation is expected, the segment will be concatenated with its neighbor and the same process is applied to the next pair of segments $(s_{i+1}, s_{i+2})$ after updating $epe(t_{i+1})$ to the predicted value. Fig.11 illustrates concatenation step for two segments and eq.(12) illustrates this step formulation and constraints where $x(s_i)$ is the current position of segment $s_i$, $s'_i$ is the new position and $\triangle x(s_i)$ is the distance that it will move to concatenate with its neighbor.

$$\begin{aligned}
&\forall(s_i, s_{i+1}) \in s \\
&s_{\min} = \text{chooseLeastEPESeg}(s_i, s_{i+1}) \\
&s_{\max} = \text{chooseMostEPESeg}(s_i, s_{i+1}) \\
&x'(s_{\min}) = x(s_{\min}) + (x(s_{\max}) - x(s_{\min})) = x(s_{\min}) + \triangle x \\
&\textbf{Subject to} \\
&\quad epe_{\mathrm{ex}}(t_{min}, \triangle x) \leq d_{\mathrm{epe}}
\end{aligned} \tag{12}$$

Algorithm1: Post OPC Processing Algorithm

```
************************ Griding***************************
$W \leftarrow$ divideRegionIntoGrids($R$,$x_{\mathrm{th}}$)
$I(T) \leftarrow$ findIntensityMap($T$)
$I(M) \leftarrow$ findIntensityMap($M$)
$G(T) \leftarrow$ applyCTR($I(M)$)
$G(M) \leftarrow$ applyCTR($I(M)$)
for each $s_i \in s$ do
    $epe_1 \leftarrow$ calculateEPE($T$, $G(T)$)
    $epe_2 \leftarrow$ calculateEPE($T$, $G(M)$)
    $\alpha[s_i] \leftarrow$ applyInterpolation($I(T,t)$, $I(M,t)$, $epe_1$, $epe_2$)
end for
************************ SRAFs Alignment and Movement***************************
for each $Sr_i \in Sr$ do
    $I_{\max} \leftarrow$ findMaxIntensityInSraf($Sr_i$)
    if $I_{\max} < I_{\mathrm{th}} - B$ then
        $Sr_i \leftarrow$ alignSraf($Sr_i$)
    end if
    $d_{\min} \leftarrow$ findDistanceToClosestNeighbor($Sr_i$, $M$)
    if $d_{\min} < d_{\mathrm{th}}$ then
        $Sr_i \leftarrow$ moveSraf($Sr_i$)
    end if
end for
$M \leftarrow$ updateMaskandCalculateMaskImage($M$, $sr$, $s$)
$Sr \leftarrow$ restoreSRafs($Sr$, $G(M)$, $T$) // restore SRAFs causing EPE violations.
$M^* \leftarrow$ updateMaskandCalculateMaskImage($M$, $sr$, $s$)
************************ Segments Alignment***************************
for each $s_i \in Ss$ do
    $epe_0 \leftarrow$ calculateEPE($T$, $G(M^*)$, $t_i$)
    $\triangle x \leftarrow$ calculateDistanceToGrid($s_i$)
    $epe_{\mathrm{ex}} \leftarrow$ calculatePredictedEPE($s_i$, $\triangle x$, $\alpha[s_i]$, $epe_0$)
    if $epe_{\mathrm{ex}} < epe_{\max}$ then
        $x(s_i) \leftarrow x(s_i) + \triangle x$
    end if
end for
$M^{**} \leftarrow$ updateMaskandCalculateMaskImage($M$, $sr$, $s$)
************************ Segments Concatenation***************************
for each $s_i \in Ss$ do
    $epe_0 \leftarrow$ calculateEPE($T$, $G(M^{**})$, $t_i$)
    $epe_1 \leftarrow$ calculateEPE($T$, $G(M^{**})$, $t_{i+1}$)
    $epe_m \leftarrow$ min($epe_0$, $epe_1$)
    $\triangle x \leftarrow$ calculateDistanceToNeighborSeg($s_i$, $s_{i+1}$)
    $s_x \leftarrow x$ chooseSegWithLeastEPE($epe_0$, $epe_1$, $s_i$, $s_{i+1}$)
    $epe_{\mathrm{ex}} \leftarrow$ calculatePredictedEPE($s_x$, $\triangle x$, $\alpha[s_x]$, $epe_m$)
    if $epe_{\mathrm{ex}} < epe_{\max}$ then
        $x(s_x) \leftarrow x(s_x) + \triangle x$
    end if
end for
************************ Output***************************
$M \leftarrow$ updateMaskandCalculateMaskImage($M$, $sr$, $s$)
```

## VI. Proposed Algorithm

Algorithm 1 illustrates our proposed post OPC algorithm whose input is target pattern $T$ and OPCed mask $M$. Mainly, our algorithm consists of 4 main steps: SRAFs alignment to improve their sizing manufacturability without being printable, SRAFs movement to make the spacing between SRAFs and other patterns within the allowable spacing, Segments alignment to eliminate small shifting distances and orthogonal undesirable small features between neighboring segments, and finally, segments concatenation to reduce the number of segments and reduce mask file size as well.

Our algorithm starts with dividing region $R$ into grid, each has $x_{\mathrm{th}} \times x_{\mathrm{th}}$ pixels area. First of all, intensity map is obtained for both $T$ and $M$ to calculate the epe to intensity rate change $\alpha(t)$ for each tap point as in eq.(10). Then, SRAFs alignment step is applied with considering the maximum intensity in each SRAF before alignment. If no printablity is expected, an SRAF will be aligned with grids borders. Thereafter, SRAFs which are violating the spacing distances are moved to satisfy this rule. This step is followed by mask image simulation and EPE recalculation to restore those SRAFs causing EPE violations in

Table 1: Edge Placement Error (EPE) VS Mask Manufacturablity

| Benchmark | #Features | Without Post OPC Processing | | | | | With Post OPC Processing | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | #EPEV | #RDRV | Data Volume | Time | Cost | #EPEV | #RDRV | Data Volume | Time | Cost | Cost (%) | #RDRV (%) |
| M1-test1 | 1030 | 5 | 491 | 12014 | 102 | 282616 | 7 | 305 | 10432 | 115 | 198047 | 29.9 | 37.8 |
| M1-test2 | 861 | 3 | 400 | 9944 | 104 | 225048 | 6 | 295 | 7873 | 113 | 185486 | 17.6 | 26.3 |
| M1-test3 | 1230 | 33 | 637 | 14861 | 102 | 498463 | 39 | 433 | 12958 | 100 | 424570 | 14.8 | 32.0 |
| M1-test4 | 296 | 0 | 249 | 4844 | 81 | 129425 | 0 | 93 | 3560 | 106 | 50160 | 61.2 | 62.7 |
| M1-test5 | 840 | 1 | 523 | 11384 | 101 | 277985 | 1 | 328 | 9452 | 102 | 178558 | 35.8 | 37.3 |
| M1-test6 | 884 | 2 | 464 | 11461 | 92 | 253553 | 6 | 283 | 9480 | 109 | 181089 | 28.6 | 39.0 |
| M1-test7 | 560 | 0 | 320 | 6643 | 85 | 166728 | 2 | 181 | 5007 | 93 | 105600 | 36.7 | 43.4 |
| M1-test8 | 422 | 0 | 259 | 5630 | 83 | 135213 | 0 | 160 | 4460 | 93 | 84553 | 37.5 | 38.2 |
| M1-test9 | 1025 | 0 | 579 | 12338 | 87 | 301925 | 2 | 381 | 10304 | 100 | 210904 | 30.1 | 34.2 |
| M1-test10 | 352 | 0 | 173 | 4201 | 82 | 90783 | 0 | 120 | 3698 | 92 | 63790 | 29.7 | 30.1 |
| **Average** | | **4.4** | **409.5** | **9332** | **91.9** | **236173.9** | **6.3** | **257.9** | **7722.4** | **103.3** | **168275.7** | **28.7** | **30.2** |
| **Ratio** | | | | | | **1.4** | | | | | **1.0** | | |

Time unit:Sec, Data Volume unit: Byte, %: Reduction Percentage
$cost = 5000 * \#EPEV + 500 * \#RDRV +$+Data Volume+Time

their neighboring tap points. The output of SRAFs alignment/ movement is mask $M^*$

Segments alignment step starts then in which segments are aligned to their closed grids as long as no EPE violations are expected as explained in eq.(8). Once this step finishes, mask $M^{**}$ image is simulated to start segments concatenation step in which each pair of segments are allowed to be concatenated together as long as no EPE violations are expected as in eq.(12). This step ends up our algorithm and post proceed mask is outputted.

## VII. Experimental Results

All experiments were executed using lithosim simulator from ICCAD 2013 CAD contest [13]. In this simulator, a target pattern $T$ is defined in $1024 \times 1024$ pixels region where each pixel represents $1nm \times 1nm$. The total number of kernels $|K| = 24$ and $I_{\mathrm{th}} = 0.225$. $x_{\mathrm{th}}$ was chosen to be $8nm$ and $d_{\mathrm{th}} = 8nm$ as well. The OPC algorithm published in [12] was used to generate OPCed masks with $10nm$ segment length. $epe_{\max} = 15nm$ as the maximum allowable EPE distance. Algorithm was executed on the public benchmarks released by IBM for the contest. For each benchmark, the total cost was calculated as in eq.(13) where $\#EPEV$ is the total EPE violations, $\#RDRV$ is the total restricted design rules violations, $V(M)$ is the mask file size, in addition to the total computation time of the entire OPC algorithm (OPC+post processing). The penalty weights are $\alpha = 5000$, and $\beta = 500$.

$$Cost(M) = \alpha * \#EPEV(M) + \beta * \#RDRV + V(M) + time \quad (13)$$

Table 1 shows our experimental results on the public benchmarks. This includes: EPE violations, Restricted Design Rule Violation Number ($\#RDRV$), mask file size, computational time, and total cost. Around $30\%$ reduction was achieved in the total cost with our post processing with no more than 15 seconds maximum increase in computation time and around only 2 additional EPE violations in average. Fig.12 illustrates a portion of benchmark 5 mask before and after post processing in which several smaller features were eliminated and the layout looks more regular.

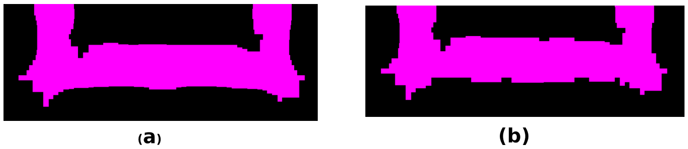

Fig. 12: Portion of Benchmark 5 Mask: (a) Before Post OPC. (b) After Post OPC.

## VIII. Conclusions and Future Work

In this paper, we proposed a post OPC processing algorithm to improve mask manufacturablity in terms of the number of design rules violations and mask data volume within a short computational time and without significant increase in EPE violations. Our experimental results show that around $30\%$ reduction in the total cost defined based on those parameters was achieved. In the future work, more cases for manufacturablity will be considered such as, hammers sizing manufacturability, segment length quantization, and Process Variability Band (PV-band) area. Furthermore, hot spots detection and fixing can be integrated with this algorithm.

## Acknowledgement

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

## References

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

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

[3] 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.

[4] 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.

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

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

[7] 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.

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

[9] 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.

[10] 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.

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

[12] 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.

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