A new algorithm of inverse lithography technology for mask complexity reduction

Li Yanghuan(李扬环)†, Shi Zheng(史峥), Geng Zhen(耿臻), Yang Yiwei(杨祎巍),
and Yan Xiaolang(严晓浪)

Institute of VLSI Design, Zhejiang University, Hangzhou 310027, China

Abstract: A new complexity penalty term called the global wavelet penalty is introduced, which evaluates the high-frequency components of masks more profoundly by applying four distinctive Haar wavelet transforms and choosing the optimal direction on which the highest frequency components of the mask will be removed. Then, a new gradient-based inverse lithography technology (ILT) algorithm is proposed, with the computation of the global wavelet penalty as the emphasis of its first phase for mask complexity reduction. Experiments with three typical 65 nm flash ROM patterns under existing 90 nm lithographic conditions show that compared with the gradient-based algorithm, which relies on the so-called local wavelet penalty, the total vertices of the three results created by the proposed algorithm can be reduced by 12.89%, 12.63% and 12.64%, respectively, while the accuracy of the lithography results remains the same.

Key words: inverse lithography technology; mask complexity; complexity penalty term; wavelet penalty

DOI: 10.1088/1674-4926/33/4/045009 **EEACC:** 2570

1. Introduction

As 193 nm wavelength photolithography systems are being pushed to fabricate devices beyond 65 nm technology, inverse lithography technology (ILT) is becoming more promising among many types of resolution enhancement technologies. Compared with optical proximity correction (OPC)[1], ILT uses a unique outcome-based approach to mathematically determine the mask patterns that produce the desired on-wafer results. A general photolithography system is represented in Eq. (1), where the function Litho(.)[2] is complicatedly nonlinear, and mainly consists of an optical model and a resist development model. The variables 'mask' and 'contour' represent the lithography mask and lithography result on wafer, respectively. The mathematical description of ILT is shown in Eq. (2).

$$\text{contour} = \text{Litho}(\text{mask}), \tag{1}$$

$$\text{mask}^* = \text{Litho}^{-1}(\text{z}), \tag{2}$$

where 'z' represents the target patterns on wafer, and mask* is the optimal mask calculated from ILT.

Various types of optimization methods can be used in ILT. Xiong and Zhang developed a simulated annealing-based method with good accuracy and fast speed[3]. Yang *et al.* described a seamless-merging-oriented parallel ILT with a gradient-based method[4]. Shen, Yu and Pan developed a DCT-2-based method and an initial sub-resolution assist feature (SRAF) insertion[5]. The level set method[6, 7] was adopted recently for its merits in handling topological complexities such as corners and cusps[8] and describing the geometric constraints[9]; its drawbacks are that some remedies should be taken to speed up the computation time[10] or deal with the so-called re-initialization issues[11]. No matter which ILT method is used, complexity reduction has always been a practical focus both for research and application[12].

A major contribution of this paper is that a new complexity penalty term called the global wavelet penalty, which is based on the Haar wavelet transform[13], is developed. Unlike the complexity penalty term in Ref. [14], which applies Haar wavelet transform on a fixed direction and is thus named as the local wavelet penalty here, the global wavelet penalty evaluates the high-frequency components of masks more profoundly by applying four distinctive Haar wavelet transforms and choosing the optimal direction on which the highest frequency components of the mask will be removed. As the global wavelet penalty is developed from the local wavelet penalty, the gradient-based ILT method used in Ref. [14] is adopted here to compare these two complexity penalty terms more conveniently. The new gradient-based ILT algorithm is proposed with the computation of global wavelet penalty as the emphasis of its first phase for mask complexity reduction. Experiments with three typical 65 nm flash ROM patterns under existing 90 nm lithographic conditions show that when comparing the ILT algorithm with the local wavelet penalty, the total vertices of the three results created by the proposed algorithm can be reduced by 12.89%, 12.63% and 12.64%, respectively, while the accuracy of the lithography results remains at the same level.

2. The Haar wavelet transform

Haar wavelet transform is a transform process that departs one-dimensional arrays or two-dimensional matrices into low- and high-frequency components. For $i \in N$, we define array $X = \{x(i)\}$, where the pixel (i.e. element) in position i of array X is $x(i)$; in the same way we can define array $A = \{a(i)\}$ and $D = \{d(i)\}$, with their pixels being expressed as in Eqs. (3) and (4).

$$a(i) = (0.5)^*[x(2i) + x(2i + 1)], \tag{3}$$

† Corresponding author. Email: liyh@vlsi.zju.edu.cn

Received 14 June 2011, revised manuscript received 28 November 2011

Table 1. The input matrix M (left) and the result matrix after Haar wavelet transform (right).

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	4	9	0	0	−1	0	0
0	0	2	4	10	8	0	0	0	8.5	10	0	0	−1.5	0	0
0	0	4	6	8	10	0	0	0	0	0	0	0	0	0	0
0	0	6	10	8	12	0	0	0	0	0	0	0	0	0	0
0	0	8	10	12	8	0	0	0	−1	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	−0.5	0	0	0	−0.5	−2	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2. The four parts of the result matrix.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	4	9	0	0	−1	0	0	0	−1	0	0	0	0	1	0
0	8.5	10	0	0	−0.5	0	0	0	−1.5	0	0	0	−0.5	−2	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Left-up part				Right-up part				Left-down part				Right-down part			

$$d(i) = (0.5) * [x(2i) - x(2i+1)]. \tag{4}$$

The process of calculating arrays A and D from array X is the Haar wavelet transform on a one-dimensional array. The pixels in arrays A and D can be considered as average values and deviation values of pixels in array X, or in other words they are low- and high-frequency components of array X, respectively.

We can see an example of Haar wavelet transform as follows. Array $X = \{x(i)\} = \{10, 13, 25, 26, 29, 21, 7, 15\}$, and after applying Haar wavelet transform on array X, the result array is $\{a(i), d(i)\}$ = {11.5, 25.5, 25, 11, –1.5, –0.5, 4, –4}. Note that arrays A and D are shown together for convenience.

Running Haar wavelet transform on a two-dimensional matrix can be considered as applying two consecutive one-dimensional Haar wavelet transforms, first on the matrix rows and then on the columns. We use matrix M at the left of Table 1 as an example to demonstrate the two-dimensional Haar wavelet transform, while the result matrix after Haar wavelet transform is shown at the right of Table 1.

The result matrix can be divided into four parts, namely left-up, right-up, left-down and right-down, all of which are shown separately in Table 2.

The above process for obtaining the four parts that compose the result matrix is Haar wavelet transform on a two-dimensional matrix. The left-up part contains the average values for the rows and columns, which possess the low-frequency components of matrix M. The other three parts can be considered as three different types of high-frequency components of matrix M: the right-up part contains deviated row and average column values; the left-down part is calculated in a reversed order compared with the right-up part; while the right-down part contains the deviation values for both matrix rows and columns.

3. The local wavelet and global wavelet penalties

Although the major target of ILT is to make wafer contour as close as possible to the ideal pattern, there are still other targets, such as a large process window and a low mask error enhancement factor. For this reason, many penalty terms are added to the gradient-based ILT cost function, as in Eq. (5).

$$\text{ILT\_COST} = \sum_{i=1}^{k} W_i \times P_i, \tag{5}$$

where k is the total number of the penalty terms, P_i represents the ith penalty term and W_i is the corresponding weight. The complexity penalty term is one of the penalty terms specifically used for mask complexity reduction. Without this complexity penalty term, a mask created by ILT would probably have complexity problems stemming from the huge increase in total vertices in the mask. The mask will become very difficult and expensive to manufacture in such a manner.

Many kinds of complexity penalty terms have been developed in earlier ILT research. As the global wavelet penalty is developed from the local wavelet penalty, we will first have a brief review on the local wavelet penalty from a previous study.

3.1. The local wavelet penalty

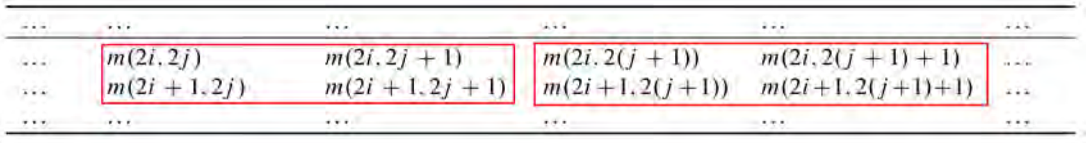
As mentioned before, the complexity penalty term proposed in Ref. [14] is named here as the local wavelet penalty (LWP), which is based on Haar wavelet transform to reduce mask complexity. Following the introduction in Section 2, the right-up part of the result matrix is now defined as matrix H; the left-down part is defined as matrix V; the right-down part of the result matrix is defined as matrix D, while the pixels in these three matrices can be expressed as in Eqs. (6)–(8), respectively. The pixel (i, j) in matrix M with a size of $U*V$ is termed as $m(i, j)$.

$$\begin{aligned} h(i,j) = {} & m(2i,2j) - m(2i,2j+1) + m(2i+1,2j) \\ & - m(2i+1,2j+1), \end{aligned} \tag{6}$$

$$\begin{aligned} v(i,j) = {} & m(2i,2j) + m(2i,2j+1) - m(2i+1,2j) \\ & - m(2i+1,2j+1), \end{aligned} \tag{7}$$

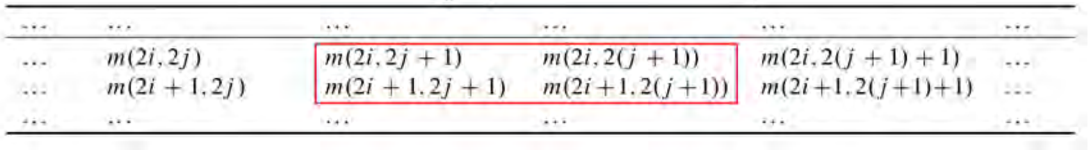
$$\begin{aligned} d(i,j) = {} & m(2i,2j) - m(2i,2j+1) - m(2i+1,2j) \\ & + m(2i+1,2j+1), \end{aligned} \tag{8}$$

Table 3. Haar groups that can be detected by Haar wavelet transform in LWP.



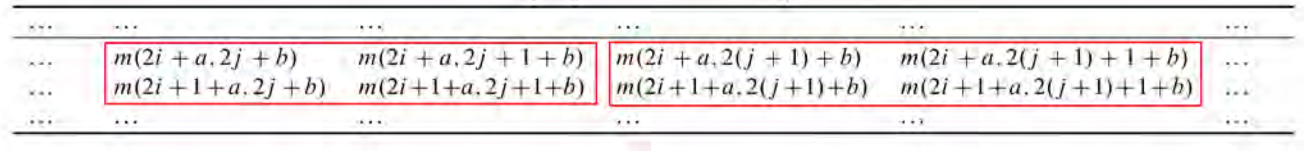
…	…	…	…	…	…
…	$m(2i,2j)$	$m(2i,2j+1)$	$m(2i,2(j+1))$	$m(2i,2(j+1)+1)$	…
…	$m(2i+1,2j)$	$m(2i+1,2j+1)$	$m(2i+1,2(j+1))$	$m(2i+1,2(j+1)+1)$	…
…	…	…	…	…	…

Table 4. The Haar groups that cannot be detected by LWP.



…	…	…	…	…	…
…	$m(2i,2j)$	$m(2i,2j+1)$	$m(2i,2(j+1))$	$m(2i,2(j+1)+1)$	…
…	$m(2i+1,2j)$	$m(2i+1,2j+1)$	$m(2i+1,2(j+1))$	$m(2i+1,2(j+1)+1)$	…
…	…	…	…	…	…

Table 5. Haar groups based on reference point $P(a,b)$.



…	…	…	…	…	…
…	$m(2i+a,2j+b)$	$m(2i+a,2j+1+b)$	$m(2i+a,2(j+1)+b)$	$m(2i+a,2(j+1)+1+b)$	…
…	$m(2i+1+a,2j+b)$	$m(2i+1+a,2j+1+b)$	$m(2i+1+a,2(j+1)+b)$	$m(2i+1+a,2(j+1)+1+b)$	…
…	…	…	…	…	…

Table 6. Four unique Haar group results based on reference points $P(0{,}0)$, $P(0{,}1)$, $P(1{,}0)$ and $P(1,1)$.

$P(a,b)$ selected	Values of a, b from $P(a,b)$	Haar group results based on reference point $P(a,b)$ with different values of i and j
$P(0,0)$	$a=0, b=0$	$m(2i,2j), m(2i,2j+1), m(2i+1,2j), m(2i+1,2j+1)$
$P(0,1)$	$a=0, b=1$	$m(2i,2j+1), m(2i,2j+2), m(2i+1,2j+1), m(2i+1,2j+2)$
$P(1,0)$	$a=1, b=0$	$m(2i+1,2j), m(2i+1,2j+1), m(2i+2,2j), m(2i+2,2j+1)$
$P(1,1)$	$a=1, b=1$	$m(2i+1,2j+1), m(2i+1,2j+2), m(2i+2,2j+1), m(2i+2,2j+2)$

where $i, j \in N$, $0 \leqslant 2i \leqslant U-2$, $0 \leqslant 2j \leqslant V-2$. LWP expressed as the sum of high-frequency components is defined in Eq. (9).

$$R_{\mathrm{w}} = \sum_{j=0}^{\frac{U-2}{2}} \sum_{i=0}^{\frac{V-2}{2}} [h(i,j)^2 + v(i,j)^2 + d(i,j)^2]. \tag{9}$$

The gradient values of LWP are given as Eq. (10).

$$\frac{\partial R_{\mathrm{w}}}{\partial m(2i+p,2j+q)} = \frac{2^*[3^*m(2i+p,2j+q) - m(2i+p,2j+q_1)}{-m(2i+p_1,2j+q) - m(2i+p_1,2j+q_1)]}, \tag{10}$$

where $p = 0$, 1 and $q = 0$, 1. $p_1 = (p+1)$ mod 2 and $q_1 = (q+1)$ mod 2. As we can see, the gradient value at certain positions actually depends on the corresponding four pixels around it.

By adding LWP into the cost function in Eq. (5), the high-frequency components of the mask could be reduced through the iterations of ILT optimization.

3.2. The global wavelet penalty

The best possible mask solution based on LWP would implicate that R_{w} in Eq. (9) is equal to zero, which can be achieved by Eq. (11).

$$m(2i,2j) = m(2i,2j+1) = m(2i+1,2j) = m(2i+1,2j+1). \tag{11}$$

We define the four pixels $m(2i,2j), m(2i,2j+1), m(2i+1,2j)$ and $m(2i+1,2j+1)$ as one Haar group. By changing the values of i and j, there are many Haar groups in matrix M, illustrated by each frame block in Table 3. As high-frequency components, or rather the pixel deviations exist in pixel groups, we can use the pixel groups to reduce the high-frequency components of the mask.

As illustrated in Table 4, high-frequency components might exist in other types of Haar group, for example $m(2i,2j+1)$, $m(2i,2(j+1))$, $m(2i+1,2j+1)$ and $m(2i+1,2(j+1))$, which cannot possibly be detected by Haar wavelet transform in LWP. This limitation of LWP could possibly lead to irregular patterns and make the ILT results not very acceptable. Based on this fact, we conclude that LWP is incomprehensive for mask complexity reduction in a certain degree and could be further improved.

As high-frequency components depend on Haar groups, we propose a new method of locating Haar groups based on reference points to detect all the possible Haar group results. $P(a,b)$ is selected as an ordinary reference point and Haar groups are formed by $m(2i+a,2j+b), m(2i+a,2j+1+b)$, $m(2i+1+a,2j+b)$ and $m(2i+1+a,2j+1+b)$ based on $P(a,b)$ with different values of i and j, where $a, b \in N$, $i, j \in Z, -a \leqslant 2i \leqslant U-2-a, -b \leqslant 2j \leqslant V-2-b$, as in Table 5.

In Appendix A, we prove that no matter how the reference point $P(a,b)$ is selected, there are only four unique Haar group results based on reference points $P(0,0)$, $P(0,1)$, $P(1,0)$ and $P(1,1)$, respectively, as in Table 6, where $i, j \in N, 0 \leqslant 2i \leqslant U-2-a, 0 \leqslant 2j \leqslant V-2-b$. We assume that these four Haar

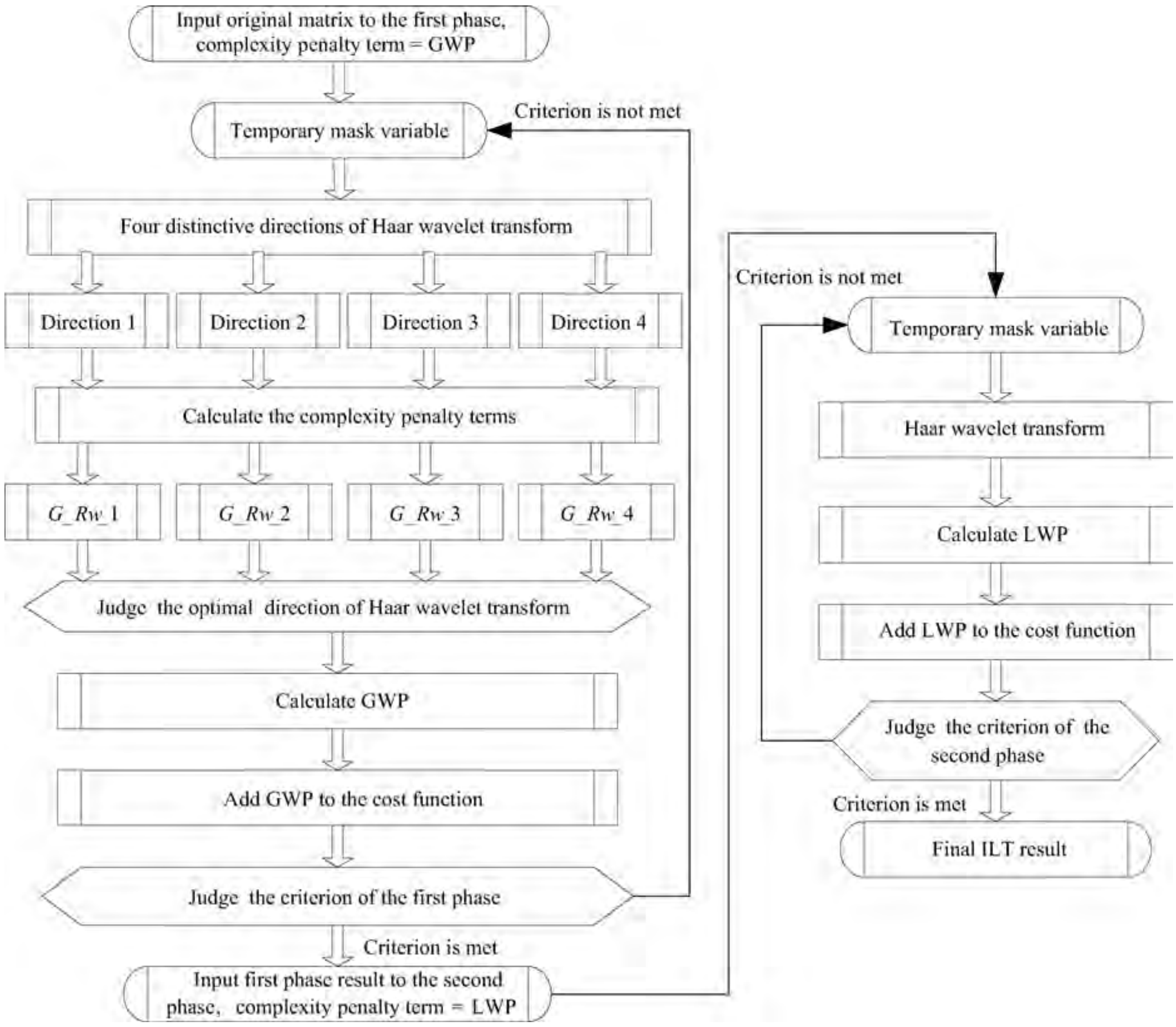

Input original matrix to the first phase, complexity penalty term = GWP
Temporary mask variable
Criterion is not met
Four distinctive directions of Haar wavelet transform
Direction 1
Direction 2
Direction 3
Direction 4
Calculate the complexity penalty terms
G_Rw_1
G_Rw_2
G_Rw_3
G_Rw_4
Judge the optimal direction of Haar wavelet transform
Calculate GWP
Add GWP to the cost function
Judge the criterion of the first phase
Criterion is met
Input first phase result to the second phase, complexity penalty term = LWP
Criterion is not met
Temporary mask variable
Haar wavelet transform
Calculate LWP
Add LWP to the cost function
Judge the criterion of the second phase
Criterion is met
Final ILT result

Fig. 1. The whole flow of our new algorithm.

group results can be detected by four distinctive directions of Haar wavelet transform, respectively, as follows:

◆ Direction 1 case: Haar group results based on reference point $P(0,0)$ can be detected by direction 1 of the Haar wavelet transform on the input matrix.

◆ Direction 2 case: Haar group results based on reference point $P(0,1)$ can be detected by direction 2 of the Haar wavelet transform on the input matrix.

◆ Direction 3 case: Haar group results based on reference point $P(1,0)$ can be detected by direction 3 of the Haar wavelet transform on the input matrix.

◆ Direction 4 case: Haar group results based on reference point $P(1,1)$ can be detected by direction 4 of the Haar wavelet transform on the input matrix.

The three different types of high-frequency components of matrix M can be calculated in Eqs. (12)–(14) based on these four Haar group results in Table 6.

$$\begin{aligned} G\_h(i,j) = {} & m(2i+a,2j+b) - m(2i+a,2j+1+b) \\ & + m(2i+1+a,2j+b) \\ & - m(2i+1+a,2j+1+b), \end{aligned} \tag{12}$$

$$\begin{aligned} G\_v(i,j) = {} & m(2i+a,2j+b) + m(2i+a,2j+1+b) \\ & - m(2i+1+a,2j+b) \\ & - m(2i+1+a,2j+1+b), \end{aligned} \tag{13}$$

$$\begin{aligned} G\_d(i,j) = {} & m(2i+a,2j+b) - m(2i+a,2j+1+b) \\ & - m(2i+1+a,2j+b) \\ & + m(2i+1+a,2j+1+b), \end{aligned} \tag{14}$$

where $i, j \in N, 0 \leqslant 2i \leqslant U-2-a, 0 \leqslant 2j \leqslant V-2-b$. The values of a and b are from $P(a,b)$ of the four direction cases. We use Eq. (15) to calculate the complexity penalty terms for the four direction cases, which are named as $G\_R_w\_1$, $G\_R_w\_2$, $G\_R_w\_3$, $G\_R_w\_4$, respectively.

$$\begin{aligned} G\_R_w\_x = {} & \sum_{j=0}^{\frac{U-2-a}{2}} \sum_{i=0}^{\frac{U-2-b}{2}} [G\_h(i,j)^2 + G\_v(i,j)^2 \\ & + G\_d(i,j)^2], \end{aligned} \tag{15}$$

where x in Eq. (15) means the direction case we select. The largest of these four complexity penalty terms is renamed as the global wavelet penalty (GWP), whose corresponding direction is chosen for applying Haar wavelet transform, as in Eq. (16).

$$R_w = \max(G\_R_w\_1, G\_R_w\_2, G\_R_w\_3, G\_R_w\_4). \quad (16)$$

The gradient values of GWP are given as Eq. (17).

$$\begin{aligned}\frac{\partial R_w}{\partial m(2i+p+a, 2j+q+b)} = \\ & 2[3m(2i+p+a, 2j+q+b) \\ & -m(2i+p+a, 2j+q_1+b) \\ & -m(2i+p_1+a, 2j+q+b) \\ & -m(2i+p_1+a, 2j+q_1+b)],\end{aligned} \quad (17)$$

where $p = 0$, 1 and $q = 0$, 1. $p_1 = (p+1)$ mod 2 and $q_1 = (q+1)$ mod 2. The values of a and b depend on the direction selection in Eq. (16). Just like Eq. (10), the gradient value of GWP at a certain position depends on the Haar group at the corresponding position. The gradient value at the place not covered by Haar groups is equal to 0. Take direction 4 as an example. As $a = 1$ and $b = 1$, the gradient at the place of row $= 0$ or column $= 0$ is equal to 0. We can find that if the direction 1 case is chosen, the gradient values of GWP in Eq. (17) are the same as the gradient values of LWP in Eq. (10), which means that LWP is a special case of GWP.

GWP in Eq. (17) is dependent on the four distinctive directions of Haar wavelet transform, and the optimal direction on which the highest frequency components of the mask will be removed is selected. While LWP in Eq. (10) is calculated by applying Haar wavelet transform in the fixed direction. By adding GWP into the cost function in Eq. (5), the high-frequency components of the mask could be reduced more effectively than in LWP.

4. The flow of our new algorithm

The whole flow of our new algorithm, which consists of two phases, is shown in Fig. 1. In the first phase GWP is selected as the complexity penalty term. When the first phase's criterion is met, its result is selected as the input of the second phase, while the complexity penalty term is changed to LWP. In this way we can incorporate both the merits of GWP and LWP in our new algorithm. The most attractive merit of GWP is that by adding GWP in Eq. (5), the high-frequency components of the mask could be reduced more effectively than LWP. However, there may be binary issues left. During the process of ILT algorithm with GWP, many pixels in a large range tend to be equal to each other, which reduces the high-frequency components at these places, and this trend may cause the pixel values at certain places to be away from 0 or 1 if some of the pixel values are originally close to 0, while others are originally close to 1, which leads to the binary issues. We use LWP in the second phase, as one important merit of LWP is that there are no binary issues for the results of the ILT algorithm with LWP. According to Eq. (11), there are only four pixels in the fixed Haar groups with a tendency to be equal to each other, which can be easily realized during the process of the ILT algorithm with LWP. To sum up, not only can the high-frequency components be reduced effectively with GWP, but binary issues can also be solved with LWP in our two-phase algorithm.

Although applying the ILT algorithm with LWP based on the first phase's result will possibly make the high-frequency components of the first phase's result increase again, we have performed lots of experiments to prove that the deviation range is very small (less than 5%) and that the high-frequency components of the final ILT result are still dominated by the first phase's result. Note that we do not combine LWP and GWP into one phase, as LWP is actually the direction 1 case of GWP. If we do so, only the high-frequency components at the Haar groups detected by the direction 1 case of GWP can be reduced effectively, while the reduction effects of the high-frequency components at the Haar groups detected by the other three direction cases of GWP may not be quite as acceptable as the influence of LWP. At the same time, some binary issues may be left in the ILT results as the influence of GWP.

5. Experiments and discussion

Before introducing the experiments, we would like to explain our study motivation. As many fabs manufacture Al-based 8 inch wafers, the minimum feature size can only reach 90 nm. It is very meaningful if we can push the feature size to 80, 70 or even 65 nm under the 90 nm lithography conditions, as the number of die in each wafer can be increased effectively. Because of the limitation of MRC rules, it is very difficult to get good OPC results based on 90 nm lithography conditions at the places with narrow spaces, such as those with 65 nm spaces between two line-ends. At the same time, ILT using a mathematical approach can produce acceptable results which cannot be realized in OPC. This merit of ILT is very meaningful, especially for producing a shrunk flash array with a minimum feature size much less than 90 nm under the 90 nm lithography conditions. It is very time consuming to run the ILT algorithm on the whole shrunk flash array, so instead we run the ILT algorithm on a shrunk standard flash unit, whose ILT result can be used to form the flash array's ILT result with certain rules. The masks created by our new algorithm have fewer complexity issues, which makes them more useful in guiding the actual production. Some adjustments will be performed based on the ILT results to produce more regular results for manufacture. For example, we will use rectangular patterns to replace the irregular SRAF produced by the ILT algorithms. Suitable adjustments should be selected to keep a good balance between the accuracy of the lithography results and the manufacturability of the final results after adjustments.

There are three different algorithms implemented and compared in our experiments: (1) an algorithm without complexity penalty term; (2) an ILT algorithm with LWP; (3) our new algorithm. For convenience, the algorithm without complexity penalty term is defined as Algorithm 1; the ILT algorithm with LWP is defined as Algorithm 2; and our new algorithm is defined as Algorithm 3. All the settings, such as main target, penalty terms and the corresponding weights, are the same except for the complexity penalty terms of these three algorithms.

We chose three 65 nm flash array results to demonstrate the validity of our new algorithm. The flash standard units, the

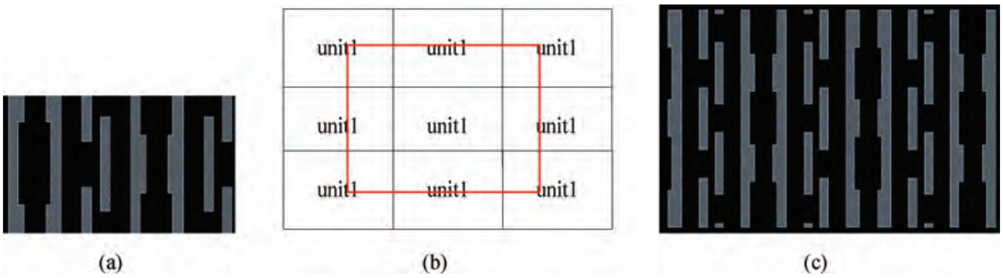


Fig. 2. (a) Flash unit 1, (b) the corresponding environment and (c) flash pattern 1.

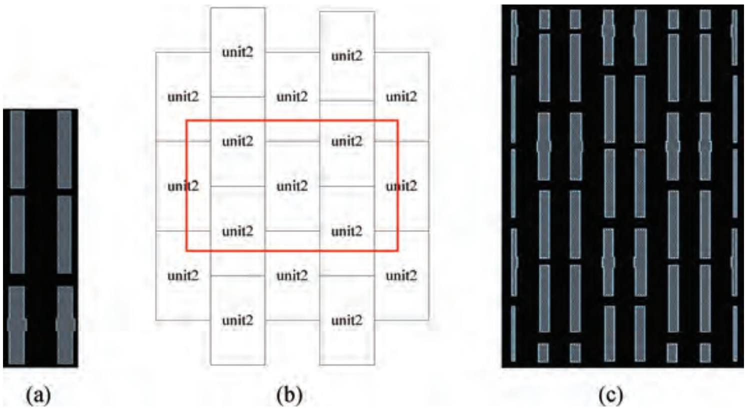


Fig. 3. (a) Flash unit 2, (b) the corresponding environment and (c) flash pattern 2.

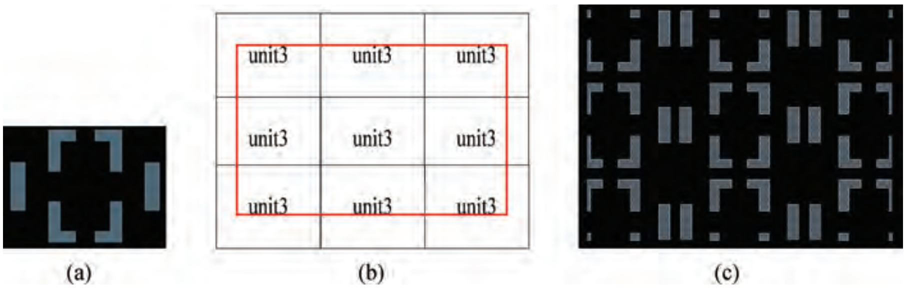


Fig. 4. (a) Flash unit 3, (b) the corresponding environment and (c) flash pattern 3.

corresponding environment and the flash patterns are shown in Figs. 2–4. The flash patterns, which are used as mask targets and original input matrices in the experiments, are formed by placing flash units in the center and symmetrically placing parts of other flash units as the environment of the center flash unit. The extending length from the other flash units to the center flash unit is longer than half of the lithography model's ambit, to guarantee the simulation accuracy of the center flash unit. Note that it is not necessary to keep symmetric relations for flash patterns, but with symmetric relations it is more convenient to apply ILT algorithms. As the environment surrounding each flash unit is composed of the same patterns, we should also update the ILT results of the flash units in the environment. Before performing the next loop, the central flash unit's ILT result is copied to replace the ILT results of the flash units in the environment. In this way, we can guarantee that all the flash units have the same ILT results.

The lithography model is a typical 90 nm model with parameters as follows: the wavelength is 193 nm, the numerical aperture is 0.7, the annular illumination with outer sigma is 0.75, the inner sigma is 0.4, the threshold is 0.3, the kernel ambit is 1280 nm, the kernel grid is 10 × 10 nm^2 and the kernel number is 8.

The cost function is shown in Eq. (18) as,

$$J(M) = \gamma_{\rm fid} F(M) + \gamma_{\rm aerial} R_{\rm aerial}(M) + \gamma_{\rm dis} R_{\rm dis}(M) + \gamma_{\rm w} R_{\rm w}(M), \tag{18}$$

where $F(M)$ stands for the main target penalty term with weight = 1, $R_{\rm aerial}(M)$ stands for aerial image penalty term with weight = 0.25, $R_{\rm dis}(M)$ stands for discretization penalty term with weight = 0.002, $R_{\rm w}(M)$ is the complexity penalty term with weight = 0.01; for Algorithm 1, $R_{\rm w}(M)$ is not set; for Algorithm 2, $R_{\rm w}(M)$ is defined in Eq. (9); and for Algorithm 3, $R_{\rm w}(M)$ is defined in Eq. (16). Except for $R_{\rm w}(M)$, detailed information of the other three penalty terms and their gradient expressions can be found in Refs. [15, 16].

As the deviations in running time for a single loop between the three algorithms are very small, we simply use loop

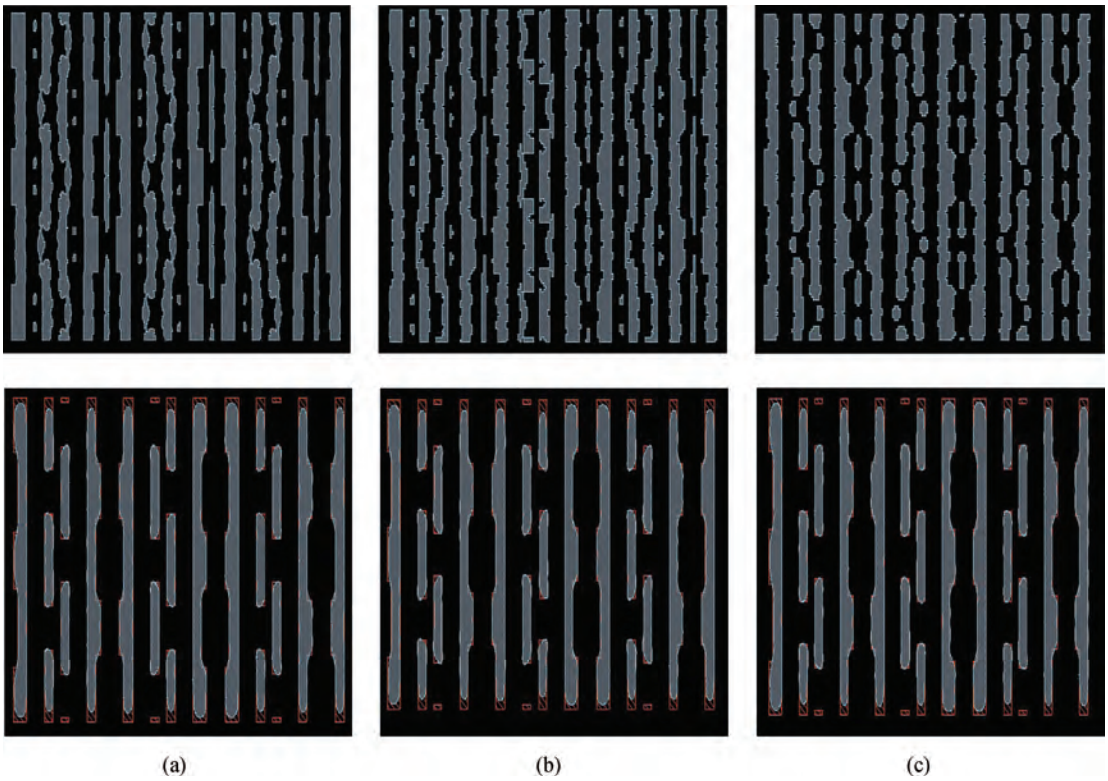


Fig. 5. The ILT results and the corresponding simulation results on flash pattern 1: (a) Algorithm 1, (b) Algorithm 2 and (c) Algorithm 3.

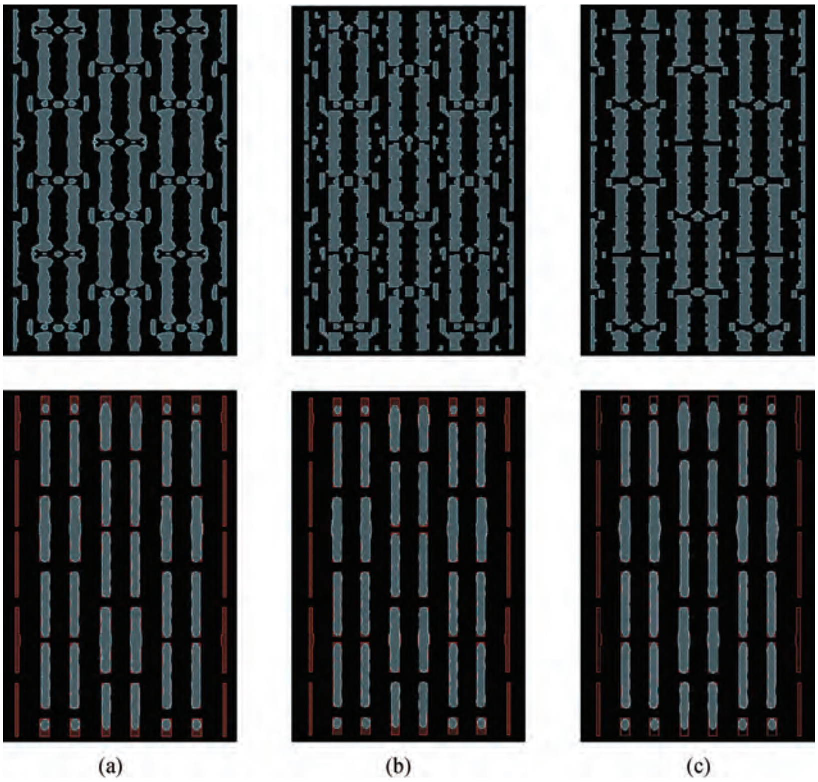


Fig. 6. The ILT results and the corresponding simulation results on flash pattern 2: (a) Algorithm 1, (b) Algorithm 2 and (c) Algorithm 3.

times to measure the running time on a 2.80 GHz computer. As the weight of $R_{\text{dis}}(M)$is the smallest compared with the other penalty terms, for Algorithm 1 and Algorithm 2 we use deviation values of $R_{\text{dis}}(M)$ for every 50 loops as the criterion, as in Eq. (19).

$$D = R_{\text{dis}}(M_{50(k+1)}) - R_{\text{dis}}(M_{50k}), \tag{19}$$

where M_{50k} stands for the temporary mask variable at loop $50k$

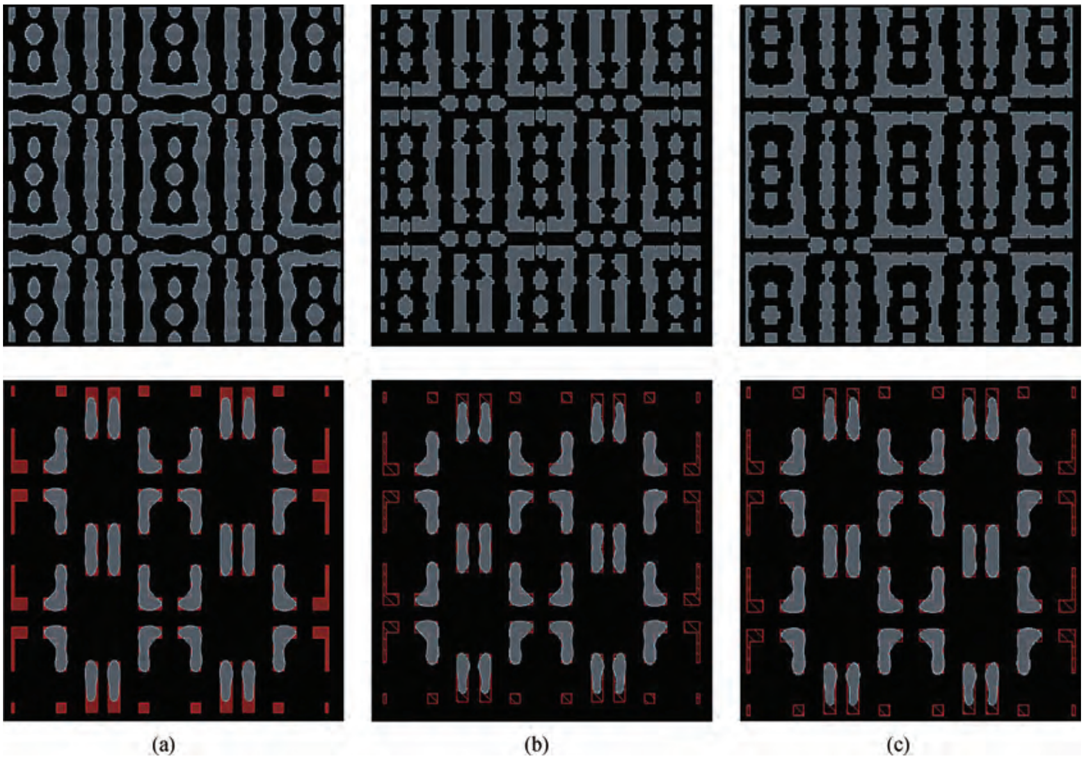


Fig. 7. The ILT results and the corresponding simulation results on flash pattern 3: (a) Algorithm 1, (b) Algorithm 2 and (c) Algorithm 3.

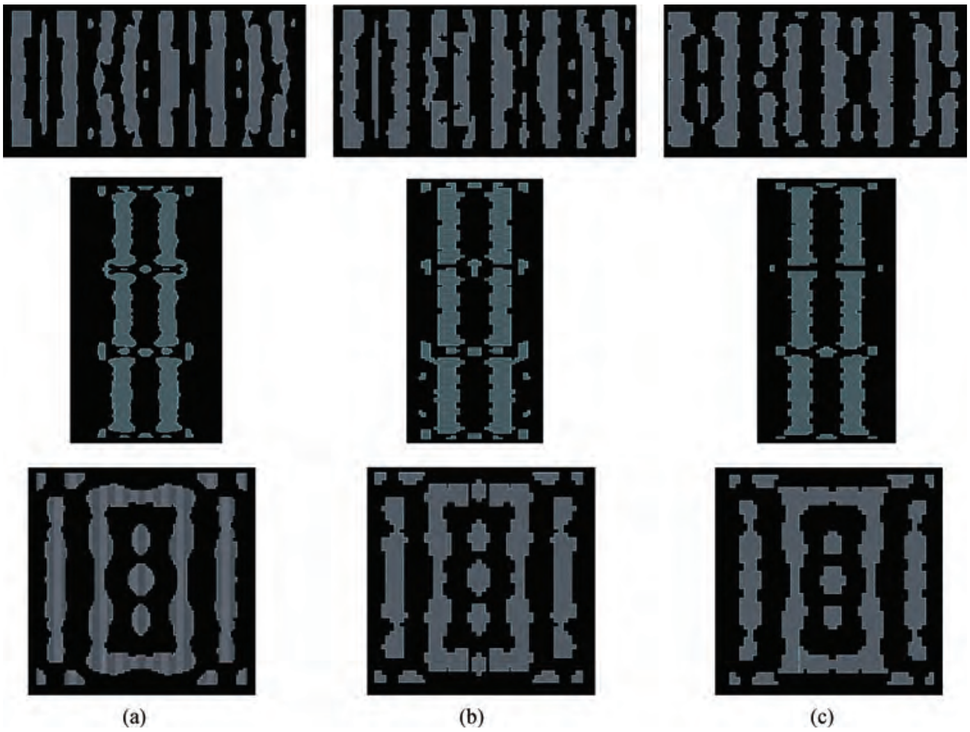


Fig. 8. The ILT results of flash unit 1 (top), 2 (middle) and 3 (down) in the center of the corresponding flash patterns: (a) Algorithm 1, (b) Algorithm 2 and (c) Algorithm 3.

in the ILT algorithm, for $k \in N$. When $k = 0$, the temporary mask variable is the same as the input mask. The expression of $R_{\text{dis}}(M)$is shown in Eq. (20).

$$R_{\text{dis}}(M) = \sum_{i=0}^{U-1} \sum_{j=0}^{V-1} [1 - (2m(i, j) - 1)^2]. \tag{20}$$

Table 7. The running time for the flash patterns after applying three algorithms.

		Algorithm 1	Algorithm 2	Algorithm 3
Loop times	Flash pattern 1	750	750	200+700
	Flash pattern 2	500	500	150+450
	Flash pattern 3	550	550	100+550

Table 8. The accuracy of the simulation results of the center flash units after applying three algorithms.

	Range of EPE	Algorithm 1	Algorithm 2	Algorithm 3
center flash unit 1	EPEs >= 5 nm	8	8	8
	The largest EPE	7.1 nm	8.3 nm	8.1 nm
center flash unit 2	EPEs >= 5 nm	0	0	0
	The largest EPE	4 nm	4.5 nm	4.7 nm
center flash unit 3	EPEs >= 5 nm	14	14	14
	The largest EPE	8 nm	8.5 nm	8.6 nm

Table 9. The total vertices of the center flash unit results after applying three algorithms.

	Algorithm 1	Algorithm 2	Algorithm 3
Total vertices of center flash unit 1 result	638	512	446
Total vertices of center flash unit 2 result	716	380	332
Total vertices of center flash unit 3 result	550	364	318

If the deviation value of 'D' in Eq. (19) is less than the specific value, we conclude that the ILT result is acceptable and the corresponding loop times are determined. The smaller the specific value we chose, the better the accuracy of the lithography results, while a greater running time of the ILT algorithm should be taken. A suitable specific value should be selected to keep a good balance between the running time of the ILT algorithm and the accuracy of the lithography results. During the experiments, we set the specific value equal to '10'. When it comes to Algorithm 3, which consists of two phases, we set two different criteria. In the first phase, as the main target is to reduce the high-frequency components of the mask effectively and $R_{\mathrm{w}}(M)$'s weight is larger than $R_{\mathrm{dis}}(M)$'s weight, we use deviation values of $R_{\mathrm{w}}(M)$ in Eq. (16) instead of deviations of $R_{\mathrm{dis}}(M)$ as the criterion to speed up our new algorithm, as in Eq. (21).

$$D = R_{\mathrm{w}}(M_{50(k+1)}) - R_{\mathrm{w}}(M_{50k}). \qquad (21)$$

In the first phase of our new algorithm, we should choose a suitable specific value to keep a good balance between the running time and the reduction effects of the high-frequency components. As in Eq. (19), the specific value in Eq. (21) is set to '10'. If the deviation value of 'D' in Eq. (21) is less than '10', we conclude that the ILT result of the first phase is acceptable and the corresponding loop times are determined. The criterion for the second phase is the same as Algorithm 1 and Algorithm 2. All the running time information is shown in Table 7. Note that the results of Algorithm 3 will be expressed as A+B, where A stands for the loop times of the first phase and B stands for the loop times of the second phase.

We also checked the edge placement errors (EPEs) on every target edge as the accuracy indicator. The flash unit in the center of the flash pattern will be checked. The places with distances from corners shorter than 20 nm are filtered to avoid corner rounding issues, while the EPE information for other places is shown in Table 8. The integer means the number of places whose EPE is within the corresponding range. As the grid size is 10 × 10 nm^2, EPEs less than 5 nm are not checked.

We used the number of total vertices to measure the complexity of the ILT result. The flash unit in the center of the flash pattern will be checked. All the information is shown in Table 9.

Figures 5–7 show the ILT results of the flash patterns and their corresponding simulation results. Figure 8 shows the ILT results of the flash units in the center of the flash patterns.

As shown in Table 7, our new algorithm consists of two phases, so it takes more time to run our new algorithm than the other two algorithms. However, we are concerned more about the complexity reduction effect here, so deviations in running time are still acceptable. We can also use more advanced machines or parallel calculations to improve the running time of our new algorithm.

From Table 8, not only is the largest EPE of our new algorithm and ILT algorithm with LWP close together, but also the number of places whose EPE is within the corresponding range are equal to each other. By this, we can conclude that the accuracy of our new algorithm and ILT algorithm with LWP is at the same level.

From Table 9, for the results of the ILT algorithm with LWP, although the total vertices are much less than the results of the ILT algorithm without complexity penalty term, there are many irregular patterns, hence the results are still not quite acceptable. When it comes to our new algorithm, compared with the ILT algorithm with LWP, the total vertices are reduced by 12.89% for flash unit 1, 12.63% for flash unit 2 and 12.64% for flash unit 3, while the patterns of the ILT results are more regular. Along with more experiments applied on other complicated mask samples, our new algorithm is further proven to reduce mask complexity effectively.

Table 10. Three expressions of Haar groups based on reference point $P(a', b')$.

Expression 1	Expression 2	Expression 3
$m(2i' + a', 2j' + b')$	$m(2i' + 2i'' + a' - 2i'', 2j' + 2j'' + b' - 2j'')$	$m(2i + a, 2j + b)$
$m(2i' + a', 2j' + b' + 1)$	$m(2i' + 2i'' + a' - 2i'', 2j' + 2j'' + b' - 2j'' + 1)$	$m(2i + a, 2j + b + 1)$
$m(2i' + a' + 1, 2j' + b')$	$m(2i' + 2i'' + a' - 2i'' + 1, 2j' + 2j'' + b' - 2j'')$	$m(2i + a + 1, 2j + b)$
$m(2i' + a' + 1, 2j' + b' + 1)$	$m(2i' + 2i'' + a' - 2i'' + 1, 2j' + 2j'' + b' - 2j'' + 1)$	$m(2i + a + 1, 2j + b + 1)$

Table 11. The results of a, b, i and j in expression 3 of Table A1 for $a = a' - 2i''$, $2i = 2i' + 2i''$, $b = b' - 2j''$, $2j = 2j' + 2j''$.

Values of a', b' from $P(a', b')$	Values of $2i''$,$2j''$ selected by ourselves	Results of a, b, i, j
$a' = 0$ or 1	$2i'' = 0$	$(a = 0, 0 \leqslant 2i \leqslant U - 2)$ or $(a = 1, 0 \leqslant 2i \leqslant U - 3)$
$b' = 0$ or 1	$2j'' = 0$	$(b = 0, 0 \leqslant 2j \leqslant V - 2)$ or $(b = 1, 0 \leqslant 2j \leqslant V - 3)$
$a' = 0$ or 1	$2i'' = 0$	$(a = 0, 0 \leqslant 2i \leqslant U - 2)$ or $(a = 1, 0 \leqslant 2i \leqslant U - 3)$
$b' >= 2$	$2j'' = b'$ or $(b' - 1)$	$(b = 0, 0 \leqslant 2j \leqslant V - 2)$ or $(b = 1, 0 \leqslant 2j \leqslant V - 3)$
$a' >= 2$	$2i'' = a'$ or $(a' - 1)$	$(a = 0, 0 \leqslant 2i \leqslant U - 2)$ or $(a = 1, 0 \leqslant 2i \leqslant U - 3)$
$b' = 0$ or 1	$2j'' = 0$	$(b = 0{,}0 \leqslant 2j \leqslant V - 2)$ or $(b = 1, 0 \leqslant 2j \leqslant V - 3)$
$a' >= 2$	$2i'' = a'$ or $(a' - 1)$	$(a = 0, 0 \leqslant 2i \leqslant U - 2)$ or $(a = 1, 0 \leqslant 2i \leqslant U - 3)$
$b' >= 2$	$2j'' = b'$ or $(b' - 1)$	$(b = 0, 0 \leqslant 2j \leqslant V - 2)$ or $(b = 1, 0 \leqslant 2j \leqslant V - 3)$

Table 12. Haar groups based on reference point $P(3, 0)$.

$m(0, 0)$	$m(0, 1)$	$m(0, 2)$	$m(0, 3)$	$m(0, 4)$	$m(0, 5)$	$m(0, 6)$
$m(1, 0)$	$m(1, 1)$	$m(1, 2)$	$m(1, 3)$	$m(1, 4)$	$m(1, 5)$	$m(1, 6)$
$m(2, 0)$	$m(2, 1)$	$m(2, 2)$	$m(2, 3)$	$m(2, 4)$	$m(2, 5)$	$m(2, 6)$
$m(3, 0)$	$m(3, 1)$	$m(3, 2)$	$m(3, 3)$	$m(3, 4)$	$m(3, 5)$	$m(3, 6)$
$m(4, 0)$	$m(4, 1)$	$m(4, 2)$	$m(4, 3)$	$m(4, 4)$	$m(4, 5)$	$m(4, 6)$

Table 13. Four unique Haar group results based on reference points P(0,0), P(0,1), P(1,0), P(1,1).

$P(a, b)$ selected	Values of a, b from $P(a, b)$	Haar group results based on reference point $P(a, b)$ with different values of i and j
$P(0, 0)$	$a = 0, b = 0$	$m(2i, 2j), m(2i, 2j + 1), m(2i + 1, 2j), m(2i + 1, 2j + 1)$
$P(0, 1)$	$a = 0, b = 1$	$m(2i, 2j + 1), m(2i, 2j + 2), m(2i + 1, 2j + 1), m(2i + 1, 2j + 2)$
$P(1, 0)$	$a = 1, b = 0$	$m(2i + 1, 2j), m(2i + 1, 2j + 1), m(2i + 2, 2j), m(2i + 2, 2j + 1)$
$P(1, 1)$	$a = 1, b = 1$	$m(2i + 1, 2j + 1), m(2i + 1, 2j + 2), m(2i + 2, 2j + 1), m(2i + 2, 2j + 2)$

6. Conclusions

In this paper, a new complexity penalty term called the global wavelet penalty is developed, and the merits of both the global wavelet penalty and the local wavelet penalty are incorporated in our new algorithm. From the experimental results, compared with the ILT algorithm with LWP, the total vertices are reduced by 12.89% for flash unit 1, 12.63% for flash unit 2 and 12.64% for flash unit 3, while the accuracy of the lithography results remain at the same level. The results therefore prove that our new algorithm is significantly effective in decreasing the complexity of the so-called ILT raw mask, which is the fundamental practical mask design for pushing existing 90 nm lithographic technology into manufacturing 65 nm or even smaller features.

Appendix A: Haar groups based on different reference points

Here we detect all the possible Haar group results based on different reference points on matrix M with size U*V. It can be proved that no matter how the reference point $P(a, b)$ is selected, there are only four unique Haar group results based on reference points $P(0, 0)$, $P(0, 1)$, $P(1, 0)$ and $P(1, 1)$, respectively.

We use $P(a', b')$ as an ordinary reference point and the Haar groups based on $P(a', b')$ are shown as expression 1 and expression 2 in Table A1, where $a', b' \in N$, $i', j' \in Z$, $-a' \leqslant 2i' \leqslant U-2-a'$,$-b' \leqslant 2j' \leqslant V-2-b'$. We rewrite expression 2 to expression 3 in Table A1, where $i'', j'' \in N$, $a = a'-2i''$, $2i = 2i' + 2i''$, $b = b' - 2j''$, $2j = 2j' + 2j''$.

We calculate the values of a, b, i and j in expression 3 of Table A1 according to the values of a' and b' from $P(a', b')$, and the values of $2i''$ and $2j''$ selected by ourselves. We take $P(0, 3)$ as an example to demonstrate how to calculate the values of a, b, i and j. Here $a' = 0$, $b' = 3$. We choose $2i'' = 0$, $2j'' = 2$, which leads to $a = a' - 2i'' = 0$, $b = b' - 2j'' = 1$. As $2i = 2i' + 2i''$, $2j = 2j' + 2j''$ while $0 \leqslant 2i' \leqslant U - 2$, $-3 \leqslant 2j' \leqslant V - 2 - 3$, we have $0 \leqslant 2i \leqslant U - 2$, $0 \leqslant 2j \leqslant V - 3$. In this way all the values of a' and b' from $P(a', b')$ are detected and the corresponding values of $2i''$ and $2j''$ are selected to calculate the results of a, b, i and j, as in Table A2.

There is an example of Haar groups based on reference point $P(3, 0)$ on a matrix with a size of 5*7. Here $a' = 3$, $b' = 0$, $U = 5$, $V = 7$. The results of a, b, i and j can be expressed as: $a = 1$, $b = 0$, $0 \leqslant 2i \leqslant 2{,}0 \leqslant 2j \leqslant 5$. We can choose $2i = 0, 2$, $2j = 0, 2, 4$, while the Haar group results based on $P(3, 0)$ are shown in Table A3.

Note that no matter how the reference point $P(a', b')$ is selected, we can select corresponding values of $2i''$ and $2j''$, which lead to four unique results of a, b, i and j in Table A2. Applying values of a, b, i and j to expression 3 of the Haar groups in Table A1, we can get four unique Haar group results based on corresponding reference points $P(0, 0)$, $P(0, 1)$, $P(1, 0)$, $P(1, 1)$, respectively, as in Table A4, where $0 \leqslant 2i \leqslant U-2-a$, $0 \leqslant 2j \leqslant V-2-b$. These four unique Haar group results are chosen to demonstrate the forming of GWP.

References

[1] Yang Yiwei, Shi Zheng, Yan Xiaolang. Model-based dynamic dissection in OPC. Journal of Semiconductors, 2008, 29(7): 1422

[2] Shen Shanhu, Shi Zheng, Xie Chunlei, et al. New modeling and optimization method suitable for UDSM lithography simulation. Journal of Semiconductors, 2007, 28(8): 1320

[3] Xiong Wei, Zhang Jinyu, Tsni Minchun, et al. An optimization strategy for 65 nm node photomasks. Journal of Computer-Aided Design & Computer Graphics, 2008, 20(5): 577

[4] Yang Yiwei, Shi Zheng, Shen Shanhu. Seamless-merging-oriented parallel inverse lithography technology. Journal of Semiconductors, 2009, 30(10): 106002

[5] Shen S, Yu P, Pan D Z. Enhanced DCT2-based inverse mask synthesis with initial SRAF insertion. Proc SPIE Photomask Technology, 2008, 7122: 712241

[6] Pang L, Liu Y, Abrams D. Inverse lithography technology (ILT) a natural solution for model-based SRAF at 45 nm and 32 nm. Proc SPIE, Photomask and Next-Generation Lithography Mask Technology XIV, 2007, 6607: 660739

[7] Shen Y, Wong N, Lam E Y. Level-set-based inverse lithography for photomask synthesis. Optics Express, 2009, 17(26): 23690

[8] Qian Yun, Zhang Yingjie. Level set methods and its application on image segmentation. Journal of Image and Graphics, 2008, 13(1): 7

[9] Xiao G, Son D H. E-beam writing time improvement for inverse lithography technology mask for full-chip. Proc SPIE, Photomask and Next-Generation Lithography Mask Technology, 2010, 7748: 77481T

[10] Chaudhury K N, Ramakrishnan K R. Stability and convergence of the level set method in computer vision. Pattern Recognition Letters, 2007: 884

[11] Wang Zhiliang, Zhou Zhewei. An improved level-set reinitialization solver. Applied Mathematics and Mechanics, 2004, 25(10): 1083

[12] Wang Guoxiong, Yan Xiaolang. Method of verification for manufacturing in sub-wavelength design. Chinese Journal of Semiconductors, 2006, 27(5): 819

[13] Li Z N, Drew M S. Fundamentals of multimedia. Beijing: China Machine Press, 2004

[14] Ma X, Arce G R. Binary mask optimization for inverse lithography with partially coherent illumination. Proc SPIE, 2008, 7140: 71401A

[15] Granik Y, Sakajiri K, Shang S. On objectives and algorithms of inverse methods in microlithography. Proc SPIE, 2006, 6349: 20076

[16] Poonawala A. Mask design for single and double exposure optical microlithography: an inverse imaging approach. PhD Thesis, Computer Engineering, University of California Santa Cruz, September 2007