The Selection and Creation of the Rules in Rules-Based Optical Proximity Correction

Rui Shi, Yici Cai, Xianlong Hong, Weimin Wu, Changqi Yang
CS. Tsinghua University, Beijing, China Email: shirui@tiger.cs.tsinghua.edu.cn

Abstract

Considering the efficiency and veracity of rules-based OPC applied to recent large-scale layout, we firstly point out the importance of the selection and creation of rules in rules-based OPC. Our discussion addresses the crucial factors in selecting and creating rules as well as how we select and create more concise and practical rules-base. Based on our ideas we suggest four primary rules and as a result we show some rule data in table. The automatic construction of the rules-base called OPCL is an important part of the whole rules-based OPC software.

I. Introduction

Recent advances in semiconductor manufacturing technology have caused critical dimensions (CD) to become shorter than the wavelength used for optical lithography. Thus more noticeable deformations are introduced by optical proximity effects (OPE) when layout patterns on a mask are transcribed to a wafer, such as line-width variation, corner rounding and line-end shortening. In order to get the desirable patterns on the wafer, optical proximity correction (OPC) is the most commonly used methodology, which aims at modifying the patterns on the mask in advance to compensate for the deformations [1][2].

Presently there are two kinds of methods for OPC: rules-based [3][4] and model-based [5][6]. Model-based OPC is done by iteratively simulating the transcribed shape on the wafer online and correcting a specific feature; thus uniform precision can be achieved at the cost of a large amount of time consumed by optics image simulation. On the contrary, rules-based OPC builds a rules-base in advance, which stores correction values for each representative feature, indexed by parameters describing the geometric information of the feature and its environment. Real corrections are carried out by identifying feature types, evaluating the geometric information parameters, looking up the rules-base, and applying the rules found. Apparently, rules-based approach is much faster and more practical for full chip correction.

Several companies have developed OPC software using the rules-based method and put forward respective rules. Paper [3] proposes a fast line-width correction system; in paper [4], three rules are suggested, which are the 1D, 1.5D and 2D rules. However, the situations these rules deal with are limited on occasion and the number of parameters in some rules is not proper. In order to make the rules-base robust and adaptable to a wide range of cases, we carefully select and construct the rules for our OPC system, which is the main contribution of this paper.

The rest of the paper is organized as follows: In Section II, we describe the problem about the OPC rules and point out the importance of the selection and creation of rules. In Section III and Section IV, we respectively discuss how to select and how to create concise and practical rules-base in detail. And in Section V we give our results and conclusion.

II. Problem Description

Rules-based OPC performs correction of each pattern in real IC layout according to the OPC rules determined in advance. The whole procedure is primarily completed by looking up the rules-base and figure operations. According to the geometric information of layout patterns, the rules are divided into several classes. The geometric information is picked up by the shape and the relative position of the correction target and its surrounding patterns. Because OPE comes mainly from the diffraction and interference of exposure light between nearby features within a definite distance, we only consider the correction target’s immediate environment and the surrounding patterns must be finite.

For each rule, a set of crucial parameters is needed to describe the geometric information of the correction target and its environment. The correction results in OPC are usually some small patterns added to or removed from the original layout patterns. So we can also use several parameters to describe the correction results. Different type of rule has different geometric information parameters and different correction result parameters. Thus the correction can be expressed as
\[ C_l = f(L, I, P_l) \]  
\[ C_l = \{C_{l1}, C_{l2}, ..., C_{ln}\} \]  
\[ L = \{\lambda, NA, defocus, \sigma, CD\} \]  
\[ I \in \{edge, corner, lineend, ...\} \]  
\[ P_l = \{P_{l1}, P_{l2}, ..., P_{lm}\} \]

\( C_l \) is the set of correction result parameters. \( L \) is the set of optical lithography parameters. Thereinto \( \lambda \) is the wavelength, \( NA \) is the numerical aperture, \( \sigma \) is the partial coherence factor, and \( CD \) is the critical dimension. \( I \) is the type of the rule. \( P_l \) is the set of geometric information parameters.

Since corrections to real layout patterns in rules-based approach are determined by looking up the rules-base, its format and content will greatly affect the efficiency and precision of the OPC process. Therefore, selection and creation of these rules are essential to the rules-based approach.

The selection and construction of our rules are based on an OPC layout optimization tool called OPCM [7], which takes layout patterns as its input and outputs the corresponding correction result. Due to the limited input pattern size and the unbearable time/space consumed by OPCM, it can not be applied directly to a full chip mask but it is suitable for the selection and creation of the rules.

### III. Selecting Rules

#### (1) Crucial Factors

A good rules-base should cover all possible instances in layout patterns. A larger rules-base usually means more precision and adaptability for the correction system. However, this also indicates more computation effort during the correction process, since a new search will be executed on the rules-base each time a layout pattern is identified to get the corresponding correction result. Thus we must control the rules-base in proper size. Two aspects should be taken into consideration.

First, the number of rule types should be carefully chosen. More detailed and complicated rule types will describe layout patterns more accurately. However, the optimization process (OPCM), the construction of the rules-base and the pick-up of the geometric information parameters will be more complicated. With the scale of circuits becoming larger and larger, the number of rule types should be kept as small as possible while maintaining high precision.

Second, the number of geometric information parameters in each rule must be appropriate. These parameters describe a correction target and its relationship with the environment. The number of these parameters is also one direct factor influencing the size of the rules-base. Proper number of parameters will enhance the precision of OPC with little additional burden on the system.

#### (2) Selecting Rule Classes

Considering the above crucial factors, we compare different patterns and combinations in some real layouts, test many patterns, and finally come up with four primary rules, which are called line-width, corner, line-end and hole respectively, as in Fig1. These four rules can cover a majority of patterns in real layouts.

![Fig1 Four rules pattern](image)

The bold real line is the correction target, and other patterns are its environment.

#### (3) Selecting Rule Parameters

According to some real layout patterns and the rules suggested in other papers, we can determine several parameters for each rule, but the number seems too much. For instance, supposing there are 6 parameters in some rule and each parameter has 10 different values, and then there will be \( 10^6 \) items in this rule table. The construction and search of this rule will obviously be time-consuming. So we try to choice comparative...
important parameters and control the number less than 3. The scale of the deformation area between the patterns on layout and on wafer to the area of the patterns on layout is regarded as the foundation. We change one parameter and can get three curves. Curve 1 represents the relationship between the parameter and the scale value before optimization, curve 2 represents the situation after optimization if this parameter is considered, and curve 3 represents the situation after optimization if this parameter is unconsidered. If curve 2 is similar to curve 3, that parameter will be ignored. Otherwise, that parameter will be one parameter in our rule. The parameters we choice for each rule are also shown in Fig1.

For example, let’s see the line-width rule. We show our line-width rule in Fig1, which has three parameters. We also show the general 1.5D edge rule suggested in paper [4] in Fig2, which has seven parameters. The simplicity of our edge rules is quite obvious. The three curves of L0, L1, and G0 are shown in Fig3, which shows that parameter L1 has little influence to correction, so that it can be ignored. What’s more, the patterns can be incised to obtain the same length lines so a uniform length parameter can be used instead of two. Through testing a series of patterns, the number of parameters in the line-width rule is reduced to three. Similar reduction process can be applied to other rules.

IV. Constructing Rules

(1) Crucial Factors

The construction of the rules-base primarily considers two parameters: the number of items in each rule table and the variation step of each parameter.
Table 1. A part line-width rule table  

<table>
<thead>
<tr>
<th>W</th>
<th>G</th>
<th>L</th>
<th>Edge Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.18</td>
<td>0.18</td>
<td>0.72</td>
<td>0.00</td>
</tr>
<tr>
<td>0.18</td>
<td>0.27</td>
<td>0.72</td>
<td>0.09</td>
</tr>
<tr>
<td>0.18</td>
<td>0.36</td>
<td>0.72</td>
<td>0.05</td>
</tr>
<tr>
<td>0.18</td>
<td>0.45</td>
<td>0.72</td>
<td>0.03</td>
</tr>
<tr>
<td>0.18</td>
<td>0.54</td>
<td>0.72</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table 2. A part corner rule table  

<table>
<thead>
<tr>
<th>W0</th>
<th>W1</th>
<th>G</th>
<th>Serif Width</th>
<th>Serif Overlap</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.18</td>
<td>0.18</td>
<td>0.18</td>
<td>0.09</td>
<td>0.015</td>
</tr>
<tr>
<td>0.27</td>
<td>0.18</td>
<td>0.18</td>
<td>0.075</td>
<td>0.006</td>
</tr>
<tr>
<td>0.36</td>
<td>0.18</td>
<td>0.18</td>
<td>0.084</td>
<td>0.006</td>
</tr>
<tr>
<td>0.45</td>
<td>0.18</td>
<td>0.18</td>
<td>0.072</td>
<td>0.01</td>
</tr>
<tr>
<td>0.54</td>
<td>0.18</td>
<td>0.18</td>
<td>0.072</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Remark: The unit used in above table is μm. The conditions of optical lithography as follows: wavelength = 248nm; numerical aperture = 0.5; defocus = 0; partial coherence factor = 0.7; minimum line width = 0.18μm. There are three parameters (W, G, L) in table 1 and three parameters (W0, W1, G0) in table 2 which are shown in Fig1. The correction (Edge Offset) in table 1 represents the offset of the target edge. The correction (Serif Width and Serif Overlap) in table 2 represents the value and position of the serif added to the target corner.

The variation step of parameters in table determines the number of items in table and the accuracy of the correction, so its value has much effect on system performance. In order to control the size of the rules-base and to get enough accuracy of correction, we must determine a suitable step value.

(2) Constructing Rules

It is well known that most of the IC designs are on the base of integer, in other words, the dimensions of all layout patterns are integral times of a definite value called delta (the minimum line width is usually twice of delta). So we can define the variation step of every parameter by delta. Thus the rule tables include enough items comparing with the real layouts. Since the OPE distance is bounded, we can confine the influence range of every parameter and thus control the size of rules-base.

Because the variation step of parameters is fixed, only the correction results are saved in rules-base and other information, such as the number of parameters in each rule, the variation step of each parameter and the variation range of each parameter, are saved in a descriptive file. While the specific value of parameters are unrecorded in rules-base. Furthermore the rule tables stored in rules-base are organized according to the preferential importance of parameters. Thus improved efficiency of looking up tables can be achieved.

V. Experiments and Results

We have implemented the above ideas in C on Sun Enterprise E450, and have built a rational rules-base. Table 1 and Table 2 show some rule tables. By applying these rules to correct layout patterns, we can get optimized layout designs. Thus we achieve the selection and creation of the rules, which are the foundation of the rules-based OPC software. The rules are more concise and practical.

Acknowledgments

The authors would like to thank Changsheng Ying, Ming Shen, and Yuan Wang for their constructive opinion and assistance with our program development.

References


[7] Changqi Yang, Xianlong Hong, Weimin Wu, Yici Cai, An Object-Based Approach to Optical Proximity Correction, in this proceeding.