Detailed Routing Algorithm with Optical Proximity Effects Constraint*

Zhou Qiang, Cai Yici†, Zhang Wei, and Hong Xianlong

(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*)

Abstract: We present a detailed routing algorithm considering the optical proximity effect. The light intensity is calculated beforehand and stored in look-up tables. These costs are used as a constraint to guide the sequential routing. The routing algorithm is based on constructed force directed Steiner tree routing to enhance routing efficiency. Experimental results on industrial benchmark circuits show that the presented routing algorithm can obtain much improvement considering optical effects short runtime.

Key words: optical proximity effect; detailed routing; Steiner tree; yield
EEACC: 7410D
CLC number: TN47 **Document code**: A **Article ID**: 0253-4177(2007)02-0189-07

1 Introduction

In the era of deep submicron manufacturing, the critical dimension of integrated circuits (IC) has become smaller than the lithographic wavelength. The unavoidable optical proximity effect (OPE) has become one of the major factors for bad performance and low yield rate of ICs[1]. The deformations in OPE include corner rounding, wide variation of line width, and the pulling back at the end of narrow lines. Although new lithographic technologies using shorter wavelengths have been reported, these techniques are still too costly and unstable to be widely used in industrial manufacture. Therefore, to reduce the distortions on wafers, optical proximity correction (OPC)[2~4] and phase shift masking (PSM)[5] techniques have been widely adopted in the industry. The OPC technique changes the original layout image by adding or subtracting fine serifs, whereas the PSM technique changes the thickness of masks in certain areas to control the phase of light through the masks.

Although these technologies have been proved with significant enhancements in subwavelength lithography, they still have some disadvantages. Rule-based OPC with a database of enhancement rules may not achieve high correction precision because of the limited data and the approximation of data values[3]. Model-based OPC using optical models needs many iterations of simulation of the optical lithography, which is quite time-consuming[4]. PSM increases the cost of mask-making. Moreover, the original layout image actually determines the feasibility of the correction that can be done in these post-layout processes. The OPC and PSM techniques can work neither effectively nor efficiently on an unfriendly routed layout that contains too many critical paths or areas. The added OPC features may sometimes further reduce the space between original ones, making the optical conditions even worse. Therefore, many researchers have been working to produce routing algorithms that are friendly to the post-layout OPC or PSM techniques. Reference [6] proposed an OPC-friendly maze routing algorithm that is formulated as a multi-constrained shortest path problem and is solved by the Lagrangian relaxation technique. Reference [7] presented a maze routing algorithm with the OPE cost calculated for each maze routing grid, and the local OPE constraint is checked in the process of maze routing. Reference [8] described gridless routing with the consideration of OPE. These algorithms have been proved to have good enhancements in lithographic quality but are not very suitable for large-scale industry production because of their time complexity.

In this paper we propose a detailed router a-

* Project supported by the National Natural Science Foundation of China (No.60476014)
† Corresponding author. Email: caiyc@tsinghua.edu.cn
Received 19 July 2006, revised manuscript received 9 October 2006

ware of OPE where the OPC cost is adopted as a constraint to guide the sequential routing process. The detailed routing algorithm is based on the constructed force directed (CFD) Steiner tree routing method[9]. The growth direction of the Steiner tree is determined by both the CFD Steiner tree algorithm and the constraint value of the OPC cost.

2 OPC technology and problems

OPC technologies compensate the deformations of lithography results by changing the shapes of the original layout features on the mask. Figure 1 is a schematic of a simple example of OPC technologies.

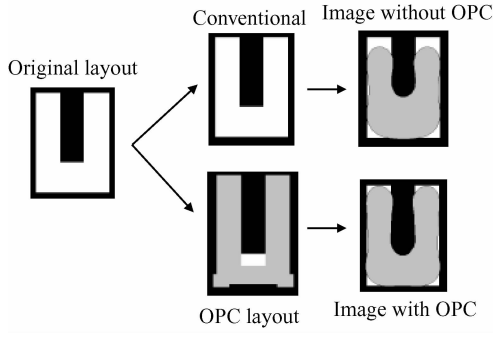

Fig. 1 Example of optical proximity correction

Currently there are two major OPC methodologies: rule-based and model-based. Rule-based OPC[2] builds the database of enhancement rules by extracting the geometrical characteristics of the original layout. Then OPC features are added onto the mask according to these pre-set rules to reduce the OPE. However, because of the geometrical measurement-based rules and the limited database, a rule-based OPC system cannot handle rules which are too complicated. This means that it is difficult for the rule-based methodology to achieve high correction precision[3].

Model-based OPC[4] simulates the optical system by iteratively applying a pre-defined optical model. Small OPC features called serifs are inserted or subtracted if the light intensity at the check points does not satisfy certain requirements. Although this methodology features better flexibility and higher precision, it requires many simulation iterations and is quite time-consuming.

Although OPC is a widely used resolution enhancement technique (RET), the calculation process is quite time-consuming. The efficiency of OPC also relies heavily on the quality of the original layout. From many available routing patterns, an unfriendly routing methodology may choose the one with the highest OPC cost and process runtime. It may even produce an OPE that cannot be corrected by post-layout OPC. Therefore, manufacturability should be considered in the routing process to enhance the production efficiency and yield.

In lithography, light diffracted from a routed net affects that from other nets. The extra light may affect other surrounding nets constructively or destructively. However, this effect decays rapidly when the distance exceeds several wavelengths. The sum of interference light intensity on a net is defined as the OPC cost of the net. The task of an OPC-friendly routing system is to constrain the OPC cost for the post-layout OPC process. Because the optical interference of the routed nets only affects other nets on the same layer, the routing system only needs to consider OPE between parallel paths if the horizontal and vertical paths are strictly separated to different layers.

3 OPC cost calculation and CFD Steiner tree based routing algorithm

3.1 Optical system model

The characteristics of the illumination light are the major factors which control the light transmission model. Typically, there are three types of illumination: coherent, incoherent, and partially coherent. Because the waves of coherent illumination are correlative in phase, their intensity of output light needs to be calculated by adding wave vectors linearly. The output light intensity of incoherent illumination is the scalar sum of the intensity from each light point, since the phases of waves are disordered. A partially coherent system can be approximated as the weighted sum of many coherent systems.

From Ref. [6], we can calculate the light intensity using the following three illumination models[6]:

· Coherent illumination:

$$I(r) = |f(r)h(r)|^2 \quad (1)$$

• Incoherent illumination:

$$I(r) = f(r)^2 \mid h(r) \mid^2 \tag{2}$$

• Partially coherent illumination:

$$I(r) = \sum_{i=1}^{n} \beta_i \mid f(r) h_i(r) \mid^2 \tag{3}$$

• The function $f(r)$ is binary:

$$f(r) = \begin{cases} 1, & \text{transparent} \\ 0, & \text{blocked} \end{cases} \tag{4}$$

We can assume that the illumination light-wave is a plane wave. Because the phase and vibration of a light-wave is the same in the same plane, the points within the layout mask can be classified into two types: transparent and blocked. Therefore, function $f(r)$ is defined as a binary function to show whether a certain point on the mask allows the light to go through, where r is the two-dimensional vector that represents the position on the plane.

The function $h(r)$ is the sine-like amplitude-impulse-response function of the optical lithography system, and β_i represents the degree of interference between each coherent illumination system. With a given optical lithography system, $h(r)$ and β_i can be calculated by the Hopkins functions[10]. $I(r)$ is the light intensity of the output image.

3.2 OPC cost calculation

The optical interference is mainly limited to an area of a few wavelengths. This means that the optical interference from a certain edge only needs to be calculated within its effective area, and the size of this effective area is determined by the amplitude-impulse-response function of the optical system. To calculate the interference on a certain edge from other routed patterns on the routing grid graph, only patterns within the effective region centered at the edge are necessary. Note that the coordinates represent the center of each routing edge. All patterns within the effective region are marked with coordinates of the left-most edge and the lengths of the patterns. The optical interference on the routing edge is the summation of the interference from all effective patterns. As long as the relative positions stay the same, all of the optical effects are equivalent.

Notice that the optical effects between two certain edges are decided by their relative positions under the same optical system. Therefore, the optical interference from all possible effective patterns can be calculated beforehand and stored in two-dimensional look-up tables. If the horizontal and vertical paths are strictly separated into different layers, the effective patterns will only include parallel straight paths. Each look-up table contains the value of the output light intensity with the corresponding pattern in the center of the effective area, as shown in Fig. 2. The light intensity used in our program is calculated by SPLAT[11]. If the optical system is symmetric along a certain axis, the size of these look-up tables can be halved.

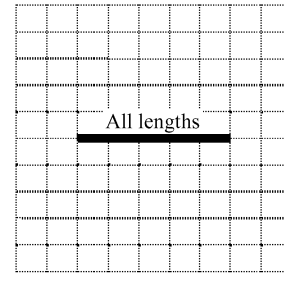

Fig. 2　Look-up table that stores the output light intensity within effective area for all lengths of patterns

For an edge e, there are two types of OPC cost. One is the optical interference from e to other paths within its effective area. Because edge e is not routed yet, the cost from e to a certain effective path is calculated as the maximum interference energy as[6]

$$\max_{g \in p_i} (T_{l(e)}(e, g)) \tag{5}$$

where p_i is the effective pattern, g is the grid point on path p_i, and $T_{l(e)}(e, g)$ is the output light intensity obtained from the look-up table (shown in Fig. 3) with edge e of length 1.

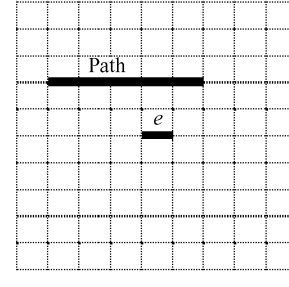

Fig. 3　Look-up table that contains light interference from a certain pattern to e within its effective area

The other kind of OPC cost is the optical interference from all other effective paths within the effective area of edge e to e itself. This ener-

gy can be calculated as the sum of the values of light interference from other effective paths, which can be looked up from tables (shown in Fig. 4) with each effective path at the center.

$$\sum T_{l(p_i)}(p_i, e) \tag{6}$$

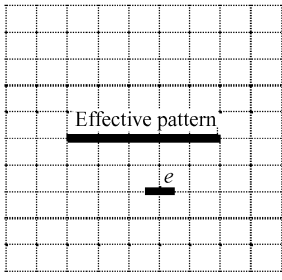

Fig. 4 Look-up table that contains light interference from e to a certain path within the effective area of e

3.3 CFD Steiner tree routing algorithm

In order to route multi-pin nets efficiently, we choose the CFD Steiner tree algorithm from many mature detailed routing methodologies as the base of our detailed router. Though Reference [12] has proved that all Steiner points can be found on Hanan grids, whose storage space is much less than track grids, the runtime cost for redrawing grids when new obstacles are added still cannot be ignored. Therefore, here we adopted the CFD Steiner tree routing proposed in Ref. [9] with track girds.

Given a track grid, we first mark the obstacle information on it. The growth vertex of a Steiner tree has four directions (up, down, left, and right) to extend on the track grids. Any direction that is blocked by obstacles is marked as "blocked", and the growth vertices of the Steiner tree are forbidden to grow in this direction. Grid points on the nearest track grid beside obstacles are blocked in the direction of the obstacle, and grid points inside an obstacle are blocked in all directions. Thus the marking problem becomes a problem of judging whether a grid point is inside the boundary of an obstacle or not. Here we adopt the scanning beam method used in graphics to solve the problem.

After the grid marking is done, the routing process begins. We route all nets in decreasing order by the largest y coordinate value of all their pins. In case of a tie, since longer nets tend to be harder to route later, we route the longest nets first.

The Steiner tree growth problem has been proved to be NP-hard, so here we define a weight middle position[9] (WMP) as the heuristic parameter to guide the process of Steiner tree growth. Assume there is a set of active nodes for a Steiner tree as $V = \{v_1, v_2, v_3, \cdots\}$, for the current growth vertex of the Steiner tree $v_i = (x_i, y_i)$. Its WMP (x_{cj}, y_{cj}) is calculated as follows:

$$d_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2 \tag{7}$$

$$D_{ij} = 1/d_{ij} \tag{8}$$

$$x_{ci} = \sum_{j \neq i, v_j \in V} (x_j D_{ij}) / \sum_{j \neq i, v_j \in V} D_{ij} \tag{9}$$

$$y_{ci} = \sum_{j \neq i, v_j \in V} (y_j D_{ij}) / \sum_{j \neq i, v_j \in V} D_{ij} \tag{10}$$

Here the weight D_{ij} is defined to make the WMP stay close to v_i. Note that the WMP may not need to be a point on the track grid. WMP just indicates the next proper growth direction.

For the current growth vertex v_i, the weight of its four possible growth directions is counted as

$$W_{\mathrm{WMP}}(d) = \alpha L(v_i, \mathrm{WMP}) \tag{11}$$

where $L(v_i, \mathrm{WMP})$ is the shortest length between v_i and its WMP on the track grid. Under the two dimensional reference frames of the track grids, it is the sum of their differences in x and y coordinates. d is the direction that needs to be calculated. Parameter α is defined as the punishing coefficient for these four directions, and its value is shown in Fig. 5.

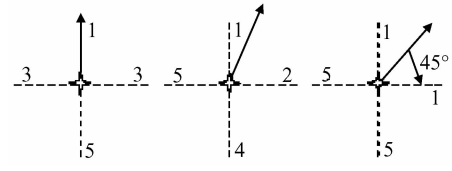

Fig. 5 Different values of α

When choosing the growth direction for the current growth vertex v_i, we select the unblocked direction with the smallest weight W_{WMP}. In case of a tie, the same direction as the growth direction chosen for v_i last time is preferred; otherwise we randomly choose one.

3.4 Detailed routing with OPC cost constraint

To constrain the optical interference, the OPC cost for each growth direction of the current growth vertex v_i is calculated. Let an edge with the unit length in the current growth direction be e and let p_i be one of the N paths within the ef-

fective area of e. Then the OPC cost from e to all other N effective paths can be calculated using Eq.(12), and the OPC cost from all other N effective paths to e can be computed by Eq.(13).

$$F_1 = \max_{g \in p_i}(T_{l(e)}(e,g)), \quad i = 1,\cdots,N \tag{12}$$

$$F_2 = \sum T_{l(p_i)}(p_i,e), \quad i = 1,\cdots,N \tag{13}$$

If one of these OPC costs F_1 or F_2 for a certain direction of the current growth vertex exceeds the constraint value C, then this direction is marked as blocked. The constraint value C is determined according to the parameters for the optical system, and C can be adjusted by the process technology from the industry.

To make the choice of next growth direction, we do not choose the unblocked direction with the smallest OPC cost, since a net grown in this way may take up necessary routing resources for nets to be routed later and may also cause further congestion. Note that as long as the OPC cost stays under the constraint value C, the routing result will be friendly for post-layout OPC processes. We can control the degree of OPC consideration by simply changing the constraint level C.

Therefore, after W_{WMP} and the OPC cost are calculated, we select the unblocked direction with the smallest W_{WMP} weight for the current growth vertex v_i. In case of a tie, the direction with the least OPC cost from other effective paths to the current growth vertex is preferred; otherwise one is randomly chosen.

4　Experimental results

The detailed router based on Steiner tree routing is implemented in C++. Experiments were done on a 1.6GHz IBM PC with 512MB memory. The parameters for the optical system are: λ = 193nm and NA = 0.5. A partially coherent optical model with parameter 0.7 is used to simulate the effective patterns and build look-up tables. The line width and space are both 140nm, which is used in the industrial 90nm process. The size of the routing grids is set at 140nm.

We have implemented the router with the assumption that horizontal and vertical edges are strictly separated in different layers. Figures 6 and 7 are simple examples of routing results by our router with and without OPC consideration. From the output light intensity, we can see that the routing quality is better for the router with OPC consideration.

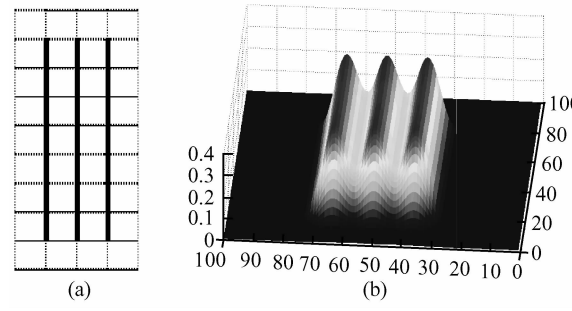

Fig.6　Routing result of example A by router without OPC consideration　(a) Grid result; (b) Output light intensity

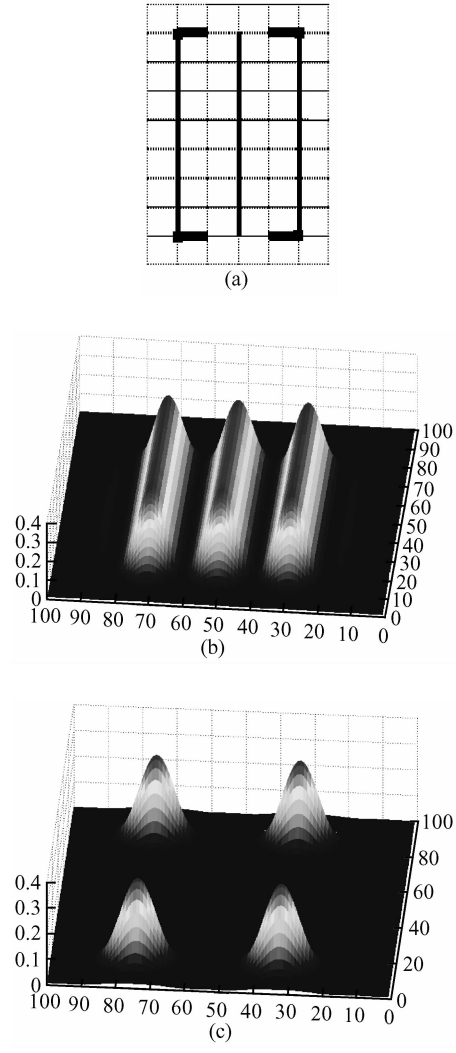

Fig.7　Routing result of example A by router with OPC consideration　(a) Grid result; (b) Output light intensity for vertical path; (c) Output light intensity for horizontal path

The benchmarks we used are industrial blocks. Table 1 describes their characteristics. BIU _ I, BIU _ II, GDC _ I, and GDC _ II are small-scale benchmarks with 2 routing layers, and full-chip routing is performed on the circuits. IBM02 and IBM09 are large-scale ones with 4 routing layers, and global routing results are fed to the detailed router.

Table 1 Characteristics of example circuits

Circuits	Circuits characteristics		
	Size/μm	Netss	Layer pairs(H-V)
GDC _ I	800×750	272	1
BIU _ I	250×800	840	1
GDC _ II	300×2100	1193	1
BIU _ II	800×800	1112	1
IBM02	82029×87528	459203	2
IBM09	134442×134456	871564	2

The routing results by detailed routers with or without OPC constraint are shown in Table 2. Though the detailed router with OPC constraint is much slower than the one without it on small-scale benchmarks, the runtime of these two kinds of routers on large-scale benchmarks is nearly the same. The runtime and the routing completion rate of the detailed router with OPC constraint on all these benchmarks are still acceptable.

The number of the OPE violation segments is the parameter that indicates the total number of segments in which the detailed router would cause OPE under our optical model. This parameter is used to estimate the final OPE violation cases of routing results of the detailed router with and without OPC consideration. If one of these OPC costs F_1 or F_2 (see also Eqs. (12) and (13)) for a certain direction of the current growth vertex exceeds the constraint value C, then this segment is an OPE violation segment. The constraint value C is determined by the parameters of the optical system.

Table 2 Characteristics of routing results

Circuits	With OPC			Without OPC		
	Run-time /s	Complete rate /%	Number of OPE violation segments	Run-time /s	Complete rate /%	Number of OPE violation segments
GDC _ I	2.4	100	0	0.70	100	85
BIU _ I	37	82	0	1.54	100	4951
GDC _ II	204	81	0	2.43	100	9277
BIU _ II	119	83	0	2.37	100	8883
IBM02	2382	93.7	0	2337	93.7	55324
IBM09	3591	96.5	0	3528	96.5	55471

For small-scale benchmarks, we can see that the violation times of benchmarks BIU _ I, BIU _ II, and GDC _ II are greater than GDC _ I. This is because GDC _ I has more routing resources, therefore the original nets are defined to be relatively more dispersed. Since the primary principle for avoiding OPE is to increase the space between close paths and let the routed paths be placed in as dispersed a configuration as possible, GDC _ I with more routing resources will result in shorter runtime and fewer OPC cost violation times.

For large-scale benchmarks, we can see that the routing completion rate is almost the same for the detailed router with or without the OPC constraint. Because the input of detailed routers is the result of global routing, the nets and resources are assigned more properly and suitably for detailed routing. Under this circumstance, we can see that the OPC constraint would not affect the routing completion rate of the detailed routing very much. From the parameter "number of OPE violation segments", we can see that using the OPC constraint can improve the OPE condition in the final routing result.

Therefore, we claim that our detailed routing algorithm with OPC constraint proves to be efficient and effective on both small-scale and large-scale industrial benchmarks.

5 Conclusion and future work

In this paper, we have presented an OPC-friendly detailed routing algorithm based on the CFD Steiner tree routing method. Two types of OPC costs are defined. One is the optical interference from the target edge to all the other paths within its effective area. The other is the optical

interference from all the other effective paths to the target edge. During sequential routing, the growth of nets is guided by the CFD Steiner tree routing method and constrained by the OPC-cost constraint value C, which can be defined by the user. Experimental results on industrial benchmarks demonstrate considerable improvement on the routing quality.

Since the routing process is sequential, the sequence of nets affects the optical proximity effect condition. A bad routing sequence may increase the complexity of calculation and therefore degrade the final quality of the routing result. Therefore, a proper management of the routing sequence will improve this methodology. We will work on this problem further in the future.

References

[1] Kahng A B, Pati Y C. Subwavelength lithography and its potential impact on design and EDA. DAC, 1999

[2] Takenouchi R, Ashia I, Kawahira H. Development of a fast line width correction system. SPIE, 2000, 4066: 688

[3] Shi Rui, Cai Yici, Hong Xianlong. The selection and creation of the rules in rules-based optical proximity correction. ASIC, 2001: 50

[4] Wang Yang, Cai Yici, Hong Xianlong, et al. Algorithm for yield driven correction of layout. ISCAS, 2004: 241

[5] Levenson M D, Viswanathan N, Simpson R A. Improving resolution in photolithography with a phase-shifting mask. IEEE Trans Electron Devices, 1982: 1828

[6] Huang L D, Wong M D F. Optical proximity correction (OPC) friendly maze routing. Proc the 41st Annual Conf on Design Automation, 2004: 186

[7] Wu Y R, Tsai M C, Wang T C. Maze routing with OPC consideration. Proc ASP-DAC Conf on Design Automation, 2005: 198

[8] Chen T C, Chang Y W. Multilevel full-chip gridless routing considering optical proximity correction. Proc ASP-DAC Conf on Design Automation, 2005: 1160

[9] Hong Xianlong. A performance-driven Steiner tree algorithm using constructed force directed approach for global routing. Chinese Journal of Semiconductors, 1995, 16(3): 218 (in Chinese) [洪先龙. 一个以时延优化为目标的力指向树 Steiner 算法. 半导体学报, 1995, 16(3): 218]

[10] Hopkins H H. On the diffraction theory of optical images. In: Microcircuit engineering, Vol. 85. Amsterdam: North-Holland Publishing Co, 1985

[11] Electronics Research Laboratory, University of California Berkeley. User's Guide for SPLAT Version 5.0

[12] Hanan M. On Steiner's problem with rectilinear distance. SIAM J of Applied Mathematics, 1996, 14(2): 255

考虑光学邻近效应的详细布线算法*

周　强　蔡懿慈†　张　为　洪先龙

（清华大学计算机科学与技术系，北京　100084）

摘要：提出了一种考虑光学邻近效应的详细布线算法. 该算法在布线过程中，充分考虑了线网走线相对位置及布线线形对其光学邻近效应的影响，通过相应的光刻模拟模型定义了用于估计光学邻近效应（optical proximity effect，OPE）的 OPE 费用函数，并采用 OPE 费用阈值控制 Steiner 树的生长方向和走线路径的选择，同时兼顾线网长度. 为提高算法效率，避免布线过程中反复调用光学模拟程序带来的算法运行速度慢的问题，对可能的走线模式建立了计算 OPE 费用所需的光强查找表格，使算法的运行速度大大提高. 在实际的工业用例上的实验结果表明，本文所提出的详细布线算法使布线结果中的 OPE 问题得到很大程度的改善，有利于后处理过程中的光学邻近效应校正技术的运用，算法的运行时间是可以接受的.

关键词：光学邻近效应校正；详细布线；Steiner 树；成品率

EEACC：7410D

中图分类号：TN47　　**文献标识码**：A　　**文章编号**：0253-4177(2007)02-0189-07

* 国家自然科学基金资助项目（批准号：60476014）

† 通信作者. Email: caiyc@tsinghua.edu.cn

2006-07-19 收到，2006-10-09 定稿