RADAR: RET-Aware Detailed Routing Using Fast Lithography Simulations

Joydeep Mitra
ECE, Univ. of Texas at Austin
& Magma Design Automation
Austin, TX 78712
jmitra@cerc.utexas.edu

Peng Yu
ECE Department
University of Texas at Austin
Austin, TX 78712
yupeng@cerc.utexas.edu

David Z. Pan
ECE Department
University of Texas at Austin
Austin, TX 78712
dpan@ece.utexas.edu

ABSTRACT

This paper attempts to reconcile the growing interdependency between nanometer lithography and physical design. We first introduce the concept of lithography hotspot, and to be more specific, edge placement error (EPE) map, then adapt efficient lithography simulation models to generate EPE maps. Less EPE usually corresponds to less RET effort and better printability later on during mask synthesis. Guided by EPE maps, we develop effective RET-aware detailed routing (RADAR) techniques that can handle full-chip capacity to enhance overall printability while maintaining other design closure. The proposed fast simulation and routing techniques are implemented in an industrial strength detailed router, and validated using some 65nm designs. Our experimental results show that we can achieve up to 40% EPE reduction with less than 5% wire length increase and reasonable CPU time.

1. INTRODUCTION

As VLSI technology continues to scale down to nanometer dimensions, the semiconductor industry is greatly challenged not only by many entangled deep submicron physical effects to reach *design closure* (for timing, signal integrity, etc.), but also by *deep sub-wavelength* lithography and other manufacturability issues to reach *manufacturing closure,* i.e., to be manufactured reliably. IC manufacturers today rely heavily on *resolution enhancement techniques* (RETs) [1][2], such as optical proximity correction (OPC), phase shift mask (PSM) and off axis illumination (OAI) to modify the chip mask database (GDSII) to achieve better printability, higher yield, and less variability. For a typical 90nm IC design process, 12 out of roughly 30 masks would require some form of resolution enhancement [2]. International Sematech recently announced its plan to adopt immersion lithography, which will further extend the 193nm lithography to 45nm node or even below. Therefore, the usage of RETs will become more pervasive for future technology generations.

With so much usage of RETs, designers can no longer leave all the burden to manufacturers since they may not be able to correct all the problems caused by the design, or to perform the correction following designer's intent [7]. Meanwhile, RETs are not cheap. They are among the primary reasons driving up the soaring mask cost (e.g., data volume increase by up to 10x, more mask write and inspection time). As silicon-manufacturing effects impact design success more heavily, there is clearly a new and growing level of *interdependency* between IC design and manufacturing.

One solution is to provide more and more rules by fabs to the design houses and CAD tools. However, as technology moves to 90nm and beyond, the number of rules quickly explodes [5-8]. This will significantly affect router performance. In addition, there may be exotic rules hard for routers to resolve. Since the rule-based models usually lack accuracy, very conservative or restricted rules may have to be given [12]. Since these restricted design rules will be applied globally, the physical layout may be overly pessimistic. On the other hand, lithography simulations, though more accurate, could be very CPU intensive. It could easily take hundreds of CPU hours to run a full chip simulation-based OPC. Our experience with PROLITH [13] and SIGMA-C [14], two leading industry lithography simulation tools, shows that it could take a few minutes to simulate a 5μm×5μm area (in accurate mode).

It will be desirable to directly link the lithography simulations with key layout optimization steps, in particular detailed routing. There is very little work so far on this topic. The work by [4] is the only attempt so far on it to our knowledge. In [4], the optical interference from neighboring edges were accumulated for the entire net under consideration. It then used maze routing with Lagrangian relaxation. It is not clear, however, how the cumulative interference relates with the final printing distortion. Also, [4] is slow as it embeds the lithography computation for each maze movement, and many iterations may be performed for Lagrangian relaxation, as it attempted a correct-by-construction approach.

In this paper, we start from an initial detailed route which has gone through, e.g., timing and congestion closure. Based on the initial routing, we introduce the concept of lithography hotspot map (LHM) to narrow the focus of fixing those regions with hotspots. To be more specific, we perform a fast lithography simulation to generate the edge placement error (EPE) to reflect lithography hotspots. Since EPE is often used by RET tools to guide the mask synthesis [9], less EPE corresponds to less RET effort. Our EPE guided correction technique naturally fits into existing design flows and is capable of handling full-chip capacity. Based on EPE maps, we propose effective detailed routing techniques to mitigate lithography hotspots. Our main contributions include the following:

- We first introduce the concept of a lithography hotspot map, which reflects the RET effort. This concept is useful for the full-chip printability improvement

- We adapt efficient lithography simulations to compute the EPE map, with fast table lookup techniques and data structures to store a ranked list of interfering neighboring edges. This information will be useful during rip-up and reroute.
- We propose effective methods including EPE guided wire spreading and ripup and reroute for post layout routing optimization. Our method requires only one full-chip litho-simulation to filter out EPE hot spots. We perform limited further simulations only when necessary in the litho hot spots. Compared to [4] which performs simulations for the entire maze routing, our approach is more suited for full-chip optimizations.
- We implement our algorithms in an industrial strength router and validate them on some real 65nm industry designs. We obtain promising results.

The rest of the paper is organized as follows. In Section 2, we review relevant background information and formulate the lithography friendly routing problem. In Section 3, we define the EPE map and describe a fast aerial image simulator that captures the first-order approximation. In Section 4, we propose EPE guided detailed routing techniques to reduce EPE. The experimental results are shown in Section 5 and we conclude in Section 6.

2. BACKGROUND AND FORMULATION

2.1 Lithography System and Modeling

For a typical lithography system simulation model as shown in Fig 1, in order to accurately model the final etch results on wafer structures, it is necessary to model both the optical component and the effects within the photoresist to get a true profile of the mask pattern within the photoresist Due to deep sub-wavelength lithographic effects, there may be significant image distortion between the original mask and the final wafer structure.

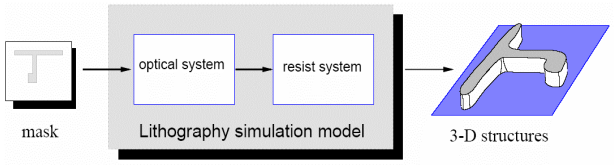


Figure 1. A typical lithography system [9].

The aerial image, as a first order approximation, is often used to estimate the final printed wafer structure. The optical system usually consists of two sets of lenses, a converging lens to generate planar waves from a partially coherent light source and an objective lens to create the aerial image of the mask pattern at the photoresist. Assuming a mask pattern of equal lines and spaces, the diffraction pattern formed at the input of the objective lens has the diffraction orders separated by a distance which is the reciprocal of the mask pattern pitch. The minimum mask pattern pitch and thus the theoretical resolution R of the imaging system is given by the equation $R = 0.5\lambda / NA$ where NA is the numerical aperture of the objective lens and λ is the wavelength. Hence the implication is that if the routing pitch is smaller than R, the aerial image and thus the placement of the routing edge on the photoresist begins to diverge from the original position in the reticle leading to unreliable etch positions on the wafer. For current and near future leading lithography systems, λ will stay at 193nm, and *NA* is roughly between 0.4 to 0.9. Thus R is bigger than the minimum 90nm or 65nm feature size. That is why several resolution enhancement techniques (RETs) have to be applied to print the fine features.

2.2 Problem Formulation

Currently RETs are applied at fabs, by a process called mask synthesis. Our goal in this paper is to reduce the post-tapeout RET effort, by modeling the lithography effects directly into the key physical layout stage. We focus this study to detailed routing as it has the required level of detailed physical geometry information for valid lithography modeling.

The lithography aware detailed routing problem is as follows:

Formulation 1: *Given an original layout (GDSII), the goal of RADAR is to improve the printability and reduce the total RET cost. The cost can be measured as the total number of assist features (line segments) added, data volume, and so on.*

To be more formal and specific for the data volume cost, for any routing edge E_i (i=1, 2, …, n, where n is the total number of detailed routing edges), let E_{ij} (j=1…m_i, with m_i being the number of new edges added by RET to print E_i properly) be the new edges in the final mask corresponding to E_i. The objective is to minimize $\sum_{i=1}^{n} m_i$ with minimum perturbation to the original design closure target.

Note that **Formulation 1** is a general formulation for an RET aware detailed router, but it may not be easy to quantify the RET cost, or it may take too much CPU to compute it during detailed routing optimization. In this paper, we will focus on another metric, edge placement error (EPE) which is easy to quantify. Given a mask pattern, the pattern edges printed on the wafer may be different from their target positions. The difference between them is called *edge placement error* (EPE). In general, the larger the EPE, the more RET effort will be necessary during mask synthesis. Thus we have the following EPE based formulation.

Formulation 2: *Given an original layout (GDSII), the goal of RADAR is to modify the routing such that the maximum and total edge placement errors are minimized.*

3. EDGE PLACEMENT ERROR MAP WITH EFFICIENT LITHO-SIMULATIONS

3.1 Edge Placement Error (EPE) Map

As described above, the *edge placement error* (EPE) for each routing edge boundary is the difference between its mask position and its final position on the wafer. This concept is used by mask synthesis and OPC tools [10,15] to iteratively adjust the assist features or move point locations such that the resulting EPE is small. However, to make EPE applicable during detailed routing, it is not practical to run OPC or lithography simulations over and over again as they are very expensive.

To make EPE useful to guide detailed routing, we introduce the concept of the *edge placement error map*. An example is shown in Fig. 2. Essentially, given an initial layout, e.g., at metal 1, we

can perform a once only, fast lithography simulation and generate the overall EPE picture for the entire chip. The EPE map can pinpoint the lithography hot spots. One can think of it as a thermal map, or a congestion map. As we shall see in Section 4, the EPE map concept fits nicely into the existing routing methodology without disturbing it drastically.

In the next subsections, we describe how to compute the EPE map effectively using a partially coherent aerial image simulation model by sampling the intensities at edge control points.

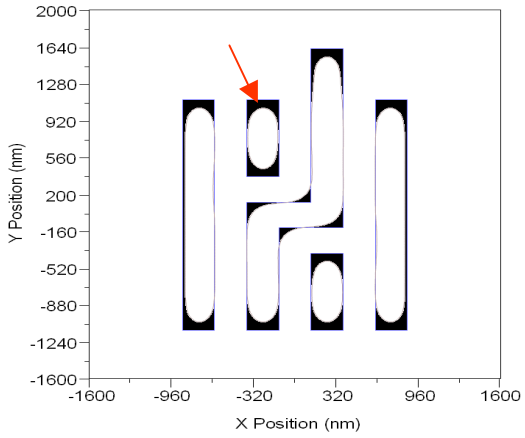


Figure 2. Edge placement error map.

3.2 Aerial Image Simulation

Every mask is composed of numerous non-overlapping rectangles. For binary masks, light passes through the rectangles completely, which means the transmission function is 1. The transmission function is 0 at all the other places. The mask transmission function F can be described by the sum of all the rectangles' transformation F_i. Here i denotes any rectangle mask pattern.

$$F(x,y)=\sum_i F_i(x,y) \tag{1}$$

For uniform fully coherent illumination, the image complex amplitude U_I can be written as the convolution of the optical system transmission function K and the mask transmission function F. (x_0,y_0) and (x_1,y_1) are the coordinates of two points on the object and image plane respectively.

$$U_I(x_1,y_1)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(x_0,y_0)K(x_1-x_0,y_1-y_0)dx_0dy_0 \tag{2}$$

Note that $K(x, y)$ will be 0 outside the region of δ_{crit} distance away from the origin. As a rule of thumb, δ_{crit} can be taken as λ/NA (see Fig. 3(a)) [11]. Eqn (2) can be reduced to the sum of convolutions as follows

$$\begin{aligned} U_I(x_1,y_1) &= \iint_{A_{(x_1,y_1)}} \sum_{i\in A_{(x_1,y_1)}} F_i(x_0,y_0)K(x_1-x_0,y_1-y_0)dx_0dy_0 \\ &= \sum_{i\in A_{(x_1,y_1)}} (F_i * K)(x_1,y_1) \end{aligned} \tag{3}$$

which can be solved by FFT (Fast Fourier Transform). The intensity at the image is $I_I=|U_I|^2$.

In real lithography systems, the sources of illumination are usually partially coherent. For partially coherent illumination, the image intensity I_I can be computed by the following equation.

$$\begin{aligned} I_I(x_1,y_1) &= \iint\iint J_0(x_0-x_0',y_0-y_0')F(x_0,y_0)F^*(x_0',y_0') \\ &\times K(x_1-x_0,y_1-y_0)K^*(x_1-x_0',y_1-y_0')dx_0dy_0dx_0'dy_0' \end{aligned} \tag{4}$$

where $J_0(x_0\text{-}x_0',y_0\text{-}y_0')$ is the mutual intensity of the light at points (x_0,y_0) and (x_0',y_0') on the mask. The superscript * denotes the complex conjugate.

The integral (4) is computationally expensive. It was shown in [10] that a partially coherent system can be approximately decomposed into a finite sum of P (< 6) fully coherent systems.

$$I_1(x,y)=\sum_{i=0}^{P-1}\Big|\sum_{j\in A_{(x,y)}}(F_j * K_i)(x,y)\Big|^2 \tag{5}$$

where K_i is the transmission function for each subsystem, F_j is the transmission function of the rectangle in region $A_{(x,y)}$. The solution of (5) is given in the next section.

3.3 Efficient Table Lookup Technique

Computing the convolutions in Eqns (3) and (5) is slow. So we pre-compute all the upper-right rectangle convolutions and retrieve them during the simulation. Fig. 3 (b) shows that the convolution for each rectangle can be decomposed into four upper-right rectangles.

$$F_p * K = F_{r_1} * K - F_{r_2} * K - F_{r_3} * K + F_{r_4} * K \tag{6}$$

We can look up $F_{rI} * K$ (i = *1, 2, 3, 4*) in the table. In this way we can compute the convolution of any arbitrary rectangle.

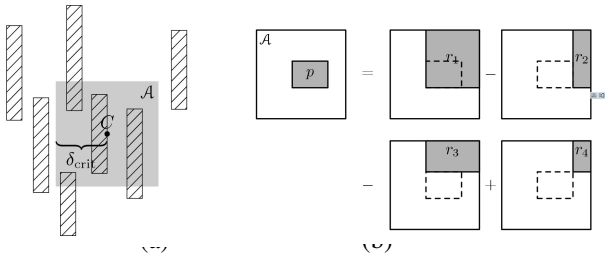


Figure 3. (a) The intensity support window. (b) Any rectangle inside the control region can be decomposed into four regions with respect to the upper right corner. Due to the linearity of the convolution, we only need to store the table w.r.t. the upper right corner.

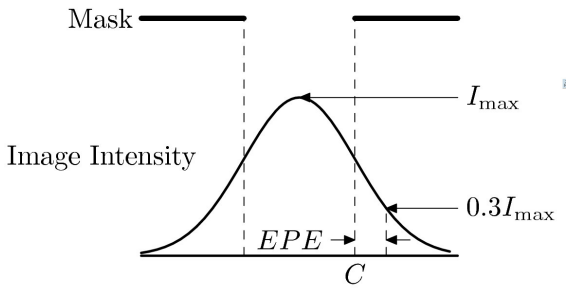


Fig 4. An approximation to compute EPE based on image density distribution

After we obtain the image intensity based on lithography simulation, a 30% rule (i.e., estimate the cutline with 0.3 I_{max}) is

used to approximate the cut line, as shown in Fig. 4. The edge placement error can be computed as the difference between the cutline and the original mask position. It can be seen from our experimental results (in Section 5.1) that our fast lithography simulations have good fidelity compared with PROLITH [13]. We will use the EPE map to guide detailed routing optimization in the next section.

4. EPE GUIDED DETAILED ROUTING

4.1 Overall Flow

Our fast lithography simulator and algorithms are incorporated into an industrial strength detailed router. The overall flow is shown in Fig. 5. The input is a detailed route after design closure. We then perform a full chip aerial image simulation. This step involves traversing all route segments layer by layer and generating control points at the route edges. A support region of rectangles for each control site is then passed to the aerial image simulator which then returns an intensity distribution around the control point which is then used to compute the cut point of the routing edge in the photoresist, as shown in Fig. 4. The difference between the cut point and original control point locations constitutes the EPE. A map of the EPEs is then available to get a visual idea of lithographic hot spots and to aid further study. We then apply a user defined EPE threshold to filter out the hotspots to be corrected. Since our simulator also provides us neighbor contributions to the EPE, we have enough information to generate routing windows and routing blockages for a rip up and reroute by the detailed router. Since the detailed router is only constrained to the intensity support region, we are assured that the new routes will not cause new OPC issues in other areas which were already below the EPE threshold. We then perform another aerial image simulation on the new routes and accept the new routes if the EPE is reduced or retain the earlier route if not.

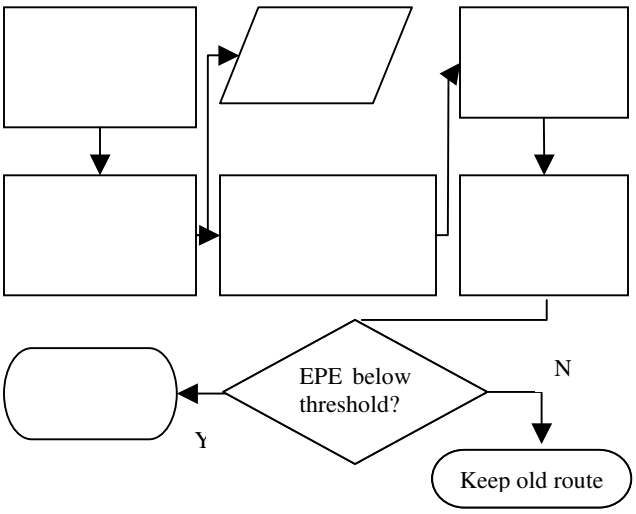


Figure 5. EPE based detailed routing flow

4.2 EPE Guided Wire Spreading

In order to compute EPE for each route segment, it is first necessary to generate a set of intensity control points for each segment. The control points were generated adaptively as shown on Fig. 6. The corners contain more dense control points as opposed to long edges, to more accurately compute corner rounding effects. Long edges contained a sparse distribution of control points with more control points being added in regions of neighbor influence within its intensity support region.

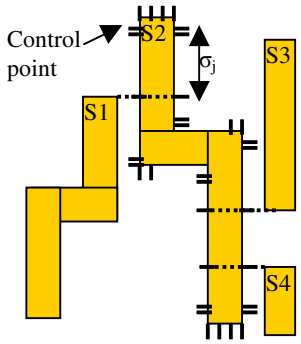


Figure 6. Adaptive control point generation to speed up the lithography simulation.

Given a simulation window W, usually a routing channel or switchbox, let $S = \{S_i : S_i \in W, i=1..n\}$ be the set of routing segments in W and $C = \{C_j : C_j \in S_i, j=1..m\}$ be the set of control points for each routing segment S_i. Moreover let ε_j be the average of the EPE at control points C_j and C_{j+1}. Then ε_j X σ_j is defined as the cumulative EPE for the rectilinear contour distance σ_j between the above control points. Now, the cumulative EPE density for the simulation window W is given by the equation

$$E_w = \sum_{i=1}^{n} \left[\sum_{j=1}^{\rho_i} \varepsilon_j \times \sigma_j \right] / L_i$$

where L_i is the contour length of segment S_i.

Our primary objective is to minimize E_w. The first proposed routing strategy is to identify all segments with EPE density E_S above a certain threshold T_S and to uniformly spread the wires within the intensity support region of the control points for the segments which are above another threshold T_C. The intuition behind this is that with wire spreading, the probability of the neighboring wires moving out of the control point's support region of influence increases. Since we break up our routing region into multiple simulation windows, we can attempt to reduce E_w defined above for each window. We first identify those windows which have E_w above a certain coarse threshold. Within each such window we try to reduce the E_S for each segment whose original E_S is above the threshold T_S. Within each such segment we then identify individual control points which have EPE above another threshold T_C. We then attempt to uniformly spread the neighbors using the following constraints: a) the neighboring segment should not enter the support region of another segment with E_S less than T_S, and b) the neighbor itself should not be one with E_S greater than T_S or a neighbor of another such segment. The reason for constraint a) is so that we do not introduce new violations through spreading, and constraint b) allows us to control circular dependencies because if the current neighbor does have E_S greater than T_S, it will be considered in a future iteration of the current simulation window. The above method is illustrated with the pseudo-code in Fig. 7.

Figure 7. Pseudo-code for EPE-map guided wire spreading

4.3 EPE Guided Ripup and Reroute (R&R)

The key observation in this strategy is the fact that the EPE computation at a control point consists of a convolution of all the rectangles within the support region of the control point, and this convolution is itself a summation of individual convolutions of each rectangle within the support region. Hence, for each control point C_j for a segment S_i, we have a set of contributing neighbors $N = \{N_k : N_k \in R_j, k=1..\alpha_j\}$ where R_j is the support region around C_j. Now assuming that the total convolution at C_j is given by U_j and the contribution to U_j by a neighbor N_k is denoted by $U(N_k)$ then the fractional contribution by neighbor N_k is defined as:

$$U_{jk} = U(N_k) / U_j$$

Let the distance from N_k to C_j be d_{jk}. Our objective is to have the detailed router increase its distance from each contributing neighbor by an amount which is a function of U_{jk}. We achieve this by generating a routing blockage of width:

$$B_{jk} = U_{jk} * d_{jk}$$

extending from the closer edge of N_k to C_j. This is depicted in Fig 8 below. Neighbor 1 has a greater contribution to the EPE than neighbor 2, hence its corresponding blockage is sized accordingly. Our second algorithm is depicted below in Fig 9. The key observation is the fact that the segment S_i under consideration will increase its distance from a specific neighbor by a detour amount which reflects the convolution influence of that neighbor on the segment control point.

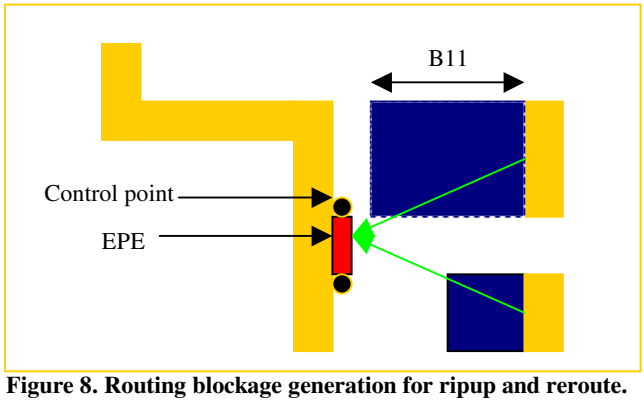


Figure 8. Routing blockage generation for ripup and reroute.

1: Sort segments with $E_S > T_S$
2: For each segment in above list (descending order of E_S)
3: Sort control points with EPE > T_C
4: Snapshot segment S
5: For each control point C (descending order of EPE)
6: For each Neighbor N in support region of C
7: Generate routing blockage
8: Ripup and reroute segment S such that constraints a) & b) are satisfied.
9: Re-simulate W.
10: If E_S has reduced for S, then accept new route else retain original route.

Figure 9. Pseudo-code for EPE-map guided ripup and reroute

5. Experimental Results

5.1. Fast Litho-simulation Validation

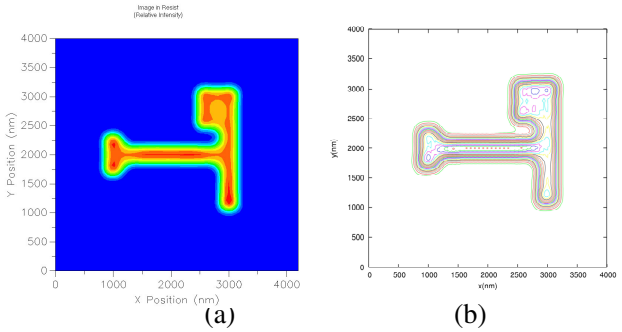


Figure 10. Aerial image comparison from (a) PROLITH (b) our fast simulation.

We also measured the CDs (Critical Dimension) of the aerial images at multiple places both in our result and in PROLITH 's result. The same mask and optical parameters are used. Our simulation is confirmed by the comparison in Table 1.

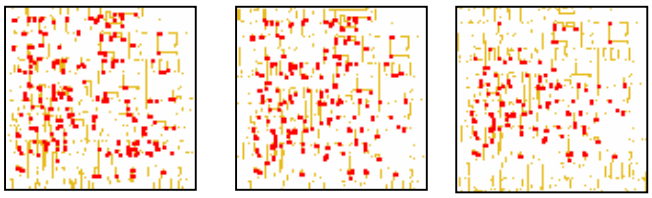
Table 1 CD comparison

Locations	Our CDs(nm)	PROLITH CDs(nm)
1	379.0	383.8
2	379.2	383.6
3	379.6	382.2
4	381.3	383.3

5.2. Routing Results

Our experiments were performed on three circuits based on 65nm design rules. The minimum metal width was 120nm and minimum metal spacing was 90nm. The manufacturing grid was 5nm. To conserve memory we used small simulation windows of size 50um X 50um and ran multiple simulations to cover an entire routing region instead of one giant simulation. For the second routing strategy using ripup and reroute it was observed that the

router tended to switch to upper layers for some of the segments under consideration. Hence our strategy was to correct the layers in increasing order, i.e. we corrected the lower layers first and fixed the routes in the lower layers before moving on to correct the upper layers. Fig 11(b) shows the actual spreading and EPE reduction within one simulation window and Fig 11(c) shows the reduction using the ripup and reroute strategy. Table 1 also shows that the wirelength increase was much lesser with R&R than with spreading.



(a) (b) (c)

Figure 11: (a) EPE before correction. (b) EPE after wire spreading. (c) EPE after ripup & reroute.

Our EPE reduction observations are summarized in Table 1. The key observation was that the ripup and reroute strategy reduced EPE by an order of 35% as compared with the EPE based wire spreading which reduced EPE by approximately 12%. The intuition behind this result is the fact that for ripup and reroute we are surgically correcting each violating segment using the neighbor convolution information thus allowing us increased correction accuracy without modifying the neighbors. On the other hand for the EPE based wire spreading method, the violating segment is untouched and we tend to uniformly spread its neighbors using more coarse objectives. EPE simulation for a block with 65K cells, in a 3000uX3000u region took approximately 135 seconds.

Table 2. EPE reduction results on three 65nm circuits

Design	Original	SP	R&R	SPwl	RRwl
ckt1	81	71	54	5%	2%
ckt2	720	612	468	20%	5%
ckt3	541	486	362	11%	5%

6. CONCLUSIONS

In this paper we present a systematic RET aware detailed routing (RADAR) technique guided by fast lithography simulations. We introduce the concept of the edge placement error (EPE) map to guide routing strategies to reduce EPE with an aim to reducing RET effort later on during the mask synthesis phase. We propose two EPE guided detailed routing strategies which are suitable for full-chip post layout optimization, one is wire spreading, and the other is a blockage based ripup and reroute strategy which shows promising results. In the future, we plan to incorporate more accurate lithography models which will include the effects of resist chemistry, exposure kinetics and post exposure bake. EPE can also be used as a framework to further study and classify routing patterns which are lithography friendly.

7. ACKNOWLEDGEMENTS

This work was partially supported by an IBM Faculty Award, an Intel Equipment Grant, and the lithography simulation software donations from KLA-Tencor and Sigma-C. We also thank Chris Mack from KLA-Tencor, Luen Heng and Ruchir Puri from IBM, Patrick McGuinness from Freescale, Prof. Martin Wong from UIUC for their helpful discussions and to Hardy Leung from Magma for his helpful suggestions and help in speeding up the simulations.

8. REFERENCES

[1] L. W. Liebmann, “Layout impact of resolution enhancement techniques: impediment or opportunity?,” in Proc. Int. Symp. on Physical Design, pp. 110–117, 2003

[2] F. M. Schellenberg, Resolution Enhancement Technology: The past, the present and extension for the future. 2004 SPIE Microlithography Symposium.

[3] K. McCullen, “Phase correct routing for alternating phase shift masks,” in Proc. Design Automation Conf., 2004.

[4] L.-D. Huang and D. F. Wong, “Optical proximity correction (OPC)-friendly maze routing,” in DAC 2004.

[5] H. K.-S. Leung, “Advanced routing in changing technology landscape,” in Proc. Int. Symp. on Physical Design, pp. 118–121, ACM Press, 2003.

[6] A. B. Kahng, “Research direcitons for coevolution of rules and routers,” in Proc. Int. Symp. on Physical Design, pp. 122–125, ACM Press, 2003.

[7] P. Gupta and A. B. Kahng, “Manufacturing-aware physical design,” in Proc. Int. Conf. on Computer Aided Design, pp. 681–687, 2003.

[8] L. Scheffer, “Physical cad changes to incorporate design for lithography and manufacturability,” in Proc. Asia and South Pacific Design Automation Conf., Jan. 2004.

[9] N. B. Cobb, Fast Optical and process Proximity Correction Algorithms for Integrated Circuit Manufacturing, Ph.D Thesis, University of California at Berkeley, 1998

[10] Y. C. Pati, A. A. Ghazanfarian, R. F. Pease, "Exploiting Structure in Fast Aerial Image Computation for IC Patterns", IEEE Trans. Semi. Mfg., Vol. 10, No. 1, pp. 62–74, Feb, 1997.

[11] Y. Liu, and A.Zakhor, "Computer Aided Phase Shift Mask Design with Reduced Complexity,” Proc. SPIE Vol. 1927, p. 477-493, Optical/Laser Microlithography VI, San Jose, March 1993.

[12] M. Lavin, F.-K. Luen, and G. Northrop, “Backend CAD Flows for ‘Restrictive Design Rules’”, ICCAD 2004

[13] PROLITH (version 8.0), KLA-Tencor Corporation.

[14] SOLID-CTM (version 6.4.1), Sigma-C Software.

[15] J. Stirniman and M.Rieger,”Fast Proximity correction with Zone sampling”,in Proceedings of SPIE Symposium on Optical Microlithography,Vol.2197,pages 294-301,Santa Clara,1994.

[16] Blast-Fusion, Magma Design Automation