RADAR: RET-Aware Detailed Routing Using Fast Lithography Simulations

Joydeep Mitra
ECE Department
University of Texas at Austin
Austin, TX 78712
joydeep@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 hotspots and the edge placement error (EPE) map to measure the overall printability and manufacturing effort. We then adapt fast lithography simulation models to generate EPE map. Guided by EPE map, we develop effective RET-aware detailed routing (RADAR) techniques that can handle full-chip capacity to enhance the overall printability while maintaining other design closure. RADAR is implemented in an industry strength detailed router, and tested using some 65nm designs. Our experimental results show that we can achieve up to 40% EPE reduction with reasonable CPU time.

Categories and Subject Descriptors

B.7.2 [Hardware]: Integrated Circuits – Design Aids

General Terms

Algorithms, Design, Experimentation, Performance, Theory

Keywords: DFM, RET, OPC, detailed routing, lithography.

1. INTRODUCTION

As VLSI technology scales down to nanometer dimension, the semiconductor industry is greatly challenged not only by many entangled *deep submicron* physical effects to reach *design closure* (such as timing, power, signal integrity), but also by *deep sub-wavelength* lithography to reach *manufacturing closure,* i.e., to be manufactured reliably. IC manufacturers nowadays rely heavily on *resolution enhancement techniques* (RET) [1][2], such as optical proximity correction (OPC) and phase shift mask (PSM) to modify chip mask database (GDSII) and achieve better printability and less variability.

These RETs, however, are not cheap. In fact, they are among the primary reasons driving up the soaring mask cost. With such pervasive usage of RETs, designers can no longer leave all burdens 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 [3]. 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 and they lack accuracy [3-6]. One may have to deal with many kinds of rules, from very restricted rules [7] to recommended design rules. Meanwhile, lithography simulations, though more accurate, could be very CPU intensive. Our experience with PROLITH [8] and SIGMA-C [9], two leading lithography simulation tools, shows that it could take a few minutes to simulate a 5μm×5μm area (in accurate mode).

Thus, it will be desirable to link fast yet reasonably accurate lithography models with the key layout optimization steps, in particular, detailed routing where exact polygons are determined. There is very little work so far on this topic. In [10], OPC-friendly maze routing was studied, where the optical interferences from neighboring edges are accumulated for an entire net under consideration. It is not clear, though, how the cumulative interference relates with the final image distortion. Also, [10] may be too slow as it embeds lithography simulation for each maze routing move, inside a Lagrangian relaxation framework.

In this paper, we propose a new RET-aware detailed routing (RADAR) framework. Our main contributions include:

- We first introduce the concept of lithography hotspot map (LHM), and to be more specific, edge placement error (EPE) map to measure the overall printability and manufacturing/RET effort. The concept is especially suitable for full-chip printability enhancement with less RET efforts.
- We adapt efficient lithography simulation techniques to compute the EPE map, with fast table lookup and data structure to store a ranked list of interfering neighboring edges.
- We propose effective RET aware routing algorithm including EPE guided wire spreading and ripup/reroute for post layout optimization. Our method requires only one full-chip litho-simulation to filter out EPE hot spots. Limited lithography re-simulations are performed only when needed. Compared to [10] which performed simulations during the entire maze routing, our approach is more suitable for full-chip optimizations.
- We implement the algorithms in an industrial strength router and validate them on some real 65nm designs. We obtain promising results.

The rest of the paper is organized as follows. In Section 2, we review relevant background and formulate the problem. In Section 3, we define the EPE map and describe a fast litho simulator that captures the first-order approximation. In Section 4, we propose EPE guided RADAR algorithm. The experimental results are shown in Section 5, followed by the conclusions in Section 6.

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

2. BACKGROUND AND FORMULATION

A typical lithography system consists of various optical and photoresist processes to print a mask image onto wafer. Due to the deep sub-wavelength lithography, there may be significant image distortions between the original mask and the final printed image. Extensive resolution enhancement techniques (RET) may be used by IC manufacturing fabs, by a step called mask synthesis. These RETs not only drive up the mask cost; they may also change timing, etc., which caused big problems for design closure.

Our goal in this paper is trying to reduce the post-tapeout RET effort, by modeling the downstream lithography effects early into the key physical layout stage, e.g., detailed routing where exact routing polygons are determined. The RET-aware detailed routing (RADAR) problem can be formulated as follows.

Formulation 1: *The goal of RADAR is to model the downstream lithography metrics into detailed routing such that the post-layout RET efforts can be minimized while maintaining design closure targets.*

Note that **Formulation 1** is generic, but it may not be easy to *quantify* the RET efforts, or it may take too much CPU to do real mask synthesis *during* detailed routing. In this paper, we focus on the post-routing optimization, and introduce the concept of edge placement error (EPE) map as a measurement of RET efforts. 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). EPE can be efficiently computed (see Section 3). Thus we have the following EPE based formulation, which is the focus of this study.

Formulation 2: *Given an original layout, the goal of RADAR is to modify the routing such that overall edge placement errors are minimized.*

3. LITHOGRAPHY METRICS & FAST SIMULATIONS

3.1 Edge Placement Error (EPE) Map

The concept of EPE is used by mask synthesis and OPC tools [11, 12] to iteratively adjust the assist features or move point locations such that the resulting EPE is small. However, it is not practical to run OPC or lithography simulations over and over again during physical design as they are very expensive.

To make EPE useful to guide detailed routing, we introduce the concept of the *edge placement error map*. Essentially, given an initial layout at any mask layer, 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 hotspots, i.e., it is a special *lithography hotspot map* (LHM). LHM is a concept similar to a thermal or congestion map, often used for thermal or routing congestion mitigation. As we shall see in Section 4, the EPE map concept fits nicely into existing routing methodology for printability optimization.

3.2 Fast Aerial Image Simulation

To efficiently generate EPE maps with reasonable fidelity, we adapt fast lithography simulation techniques. The aerial image after the optical system, as the first order approximation, is often used to estimate the final printed image. An ordinary binary mask consists of numerous non-overlapping clear and opaque rectangle regions. The so-called transmission function is 1 over clear regions and 0 over opaque regions. The overall mask transmission function F is the sum of F_i (i denotes the *i-th* clear rectangle).

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

It is common to use partially coherent illumination in today's lithography system. The aerial image intensity I_1 of a partially coherent system can be approximately decomposed into a small number of P fully coherent systems [11]

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

where K_i is the transfer function for the *i-th* fully coherent optical subsystem, F_j is the transmission function of the *j-th* rectangle in region $A_{(x,y)}$, the intensity *support region* of control point C at location *(x, y)*. The size of the support region depends on the wavelength and numerical aperture of the optical system, but in general is pretty small (1-4um). Due to the linear property, the convolution for any arbitrary rectangle inside the support region can be decomposed into four upper-right rectangles [13], as shown in Fig. 1.

$$F_p * K = F_{r_1} * K - F_{r_2} * K - F_{r_3} * K + F_{r_4} * K \tag{3}$$

We can pre-compute all upper-right rectangle convolutions and store them in a table. Then any $F_p * K$ can be efficiently computed by the table lookup as shown in Fig. 1 (b).

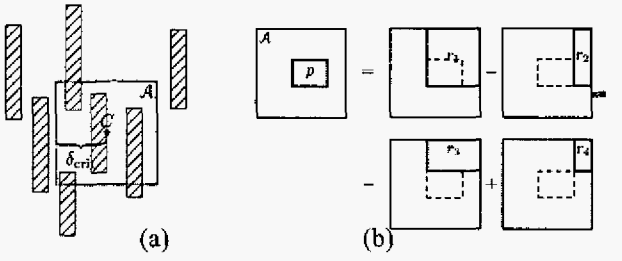

Figure 1: (a) Litho support region A for a control point C; (b) decomposition of an arbitrary rectangle p into 4 upper right corner rectangles for efficient convolution table-lookup.

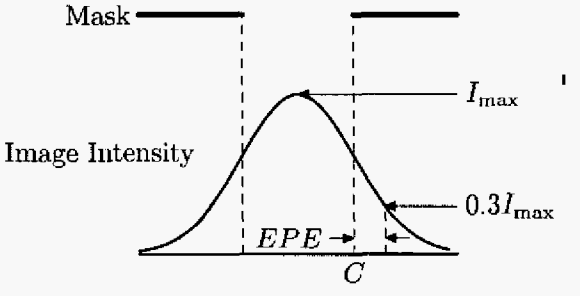

Figure 2: EPE estimation based on image intensity.

Once the image intensity distribution around the control point C is computed, its edge position can be approximated by a rule-of-thumb (30% rule) as shown in Fig. 2. The difference between the printed edge position and the original mask edge position is the edge placement error (EPE). Our experimental results (in Section 5.1) show that our fast lithography simulations have good *fidelity* compared with PROLITH [12].

4. EPE GUIDED DETAILED ROUTING

4.1 Overall Flow

The overall flow of our lithography-aware detailed routing (RADAR) is shown in Fig. 3. Given an initial routing (e.g., after design closure), we first perform a full chip fast lithography simulation to generate the EPE map. This step traverses all routing segments layer by layer and adaptively generates control points (i.e., points that potentially have printability problem) at routing segment edges. A support region for each control point is then passed to the fast aerial image simulator which returns an intensity distribution around the control point to estimate the edge placement error. We then apply a user defined EPE threshold to filter out the EPE hotspots to be corrected. Our simulator also keeps track of all neighbor contributions to EPE, which can be used to generate routing windows and routing blockages for effective EPE guided wire spreading and rip-up/reroute (R&R) by the detailed router. We perform aerial image re-simulation only when necessary (i.e., on the new routes) and accept the new routes if the EPE is reduced. Such procedure can be iterated for all lithography hotspots until there is no further improvement.

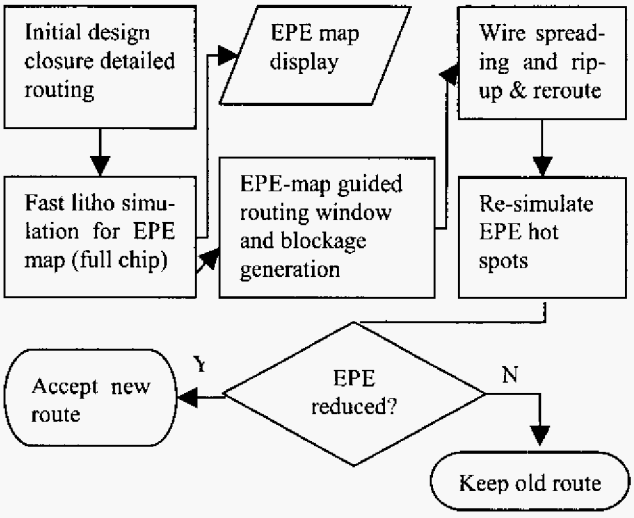

Figure 3: EPE based detailed routing flow (RADAR).

4.2 EPE Guided Wire Spreading

In order to compute EPE for each route segment, we first generate a set of intensity control points for each routing segment. The control points are generated adaptively with more dense control points near the corners than at the middle of long edges, to more accurately capture the corner rounding effects. Long edges usually contain a sparse distribution of control points, but for those locations with high neighborhood interference, more control points will be added. Our routing strategy is to identify the most critical EPE hotspots then perform the wire spreading and R&R.

Given a routing window W, such as 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 $C = \{C_{ij} : C_{ij} \in S_i, j=1...m\}$ be the set of control points for each routing segment S_i. Moreover, let ε_{ij} be the average EPE for two adjacent control points C_{ij} and $C_{i,j+1}$ and σ_{ij} be the distance between them. Then $\varepsilon_{ij} \times \sigma_{ij}$ can be used to estimate the cumulative EPE between C_{ij} and $C_{i,j+1}$. Thus, the cumulative EPE density for the simulation window W can be written as

$$E_w = \sum_{i=1}^{n} \left[\sum_{j=1}^{m} \varepsilon_{ij} \times \sigma_{ij} \right] / L_i$$

where L_i is the contour length of segment S_i.

We first break up an entire chip into uniform routing windows W and sort them based on E_w. The EPE-guided wire spreading will attempt to correct those routing windows in a sorted order (bigger E_w first) for all windows with E_w above certain threshold T_w (threshold for routing window).

Within each window, it has many routing segments. We try to reduce EPE for each segment i with E_i above certain threshold T_s (threshold for segment). Note that different control points in this segment may contribute differently to its overall EPE. Thus, we will narrow down the control points with bigger EPE (above another threshold T_C, threshold for control points). We uniformly spread the neighboring wires around those bad control points using the following constraints:

(a) The neighbor segment should not enter the support region of another segment with E_S less than T_S such that no new violations are created during spreading;

(b) The neighbor itself should not be the one with E_S greater than T_S to avoid control region circular dependency.

4.3 EPE Guided Ripup and Reroute (R&R)

The EPE computation at a control point consists of a summation of convolution of all rectangles within its support region (see Section 3). Hence, for each control point C_j of a segment S_i (the index i is dropped in C_{ij} for simplicity), we can store a ranked list of contributing neighbors $N = \{N_k : N_k \in A_j\}$, where A_j is the support region around C_j. Assuming 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 $U_j(N_k)$, the contribution by neighbor N_k to the control point C_j:

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

Let the distance from N_k to C_j be d_{jk}. To effectively reduce EPE caused by its neighbor interference, we create a routing blockage of width, $B_{jk} = U_{jk} * d_{jk}$ from N_k to C_j, then perform the rip-up and reroute. The blockage generation is illustrated in Fig. 4. Since neighbor N1 has a greater contribution to the EPE of S1 than neighbor N2, it has a bigger size blockage B11.

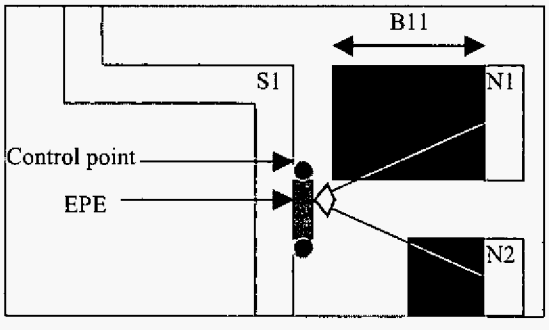

Figure 4: Routing blockage generation for ripup and reroute.

5. Experimental Results

5.1. Fast Litho-Simulation Validation

We compare our fast litho simulations with PROLITH [8] and show good fidelity (Fig. 5). The CD error is less than 1% between PROLITH and our fast lithography simulator.

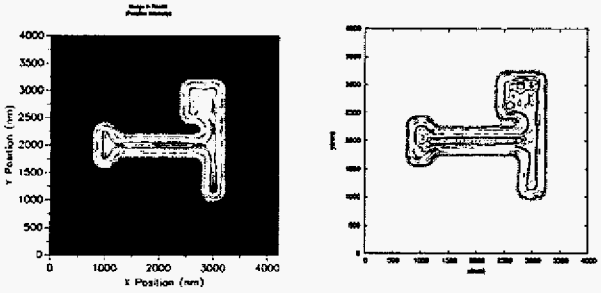

Figure 5: Comparison of PROLITH and our fast simulation.

5.2. Routing Results

Our experiments are performed on three industry circuits based on 65nm design rules, using an industry strength router [14]. The minimum metal width/spacing is 120nm/90nm, respectively. The manufacturing grid is 5nm. To conserve memory, we use routing windows of size 50um x 50um and run multiple simulations to cover the entire routing region. Fig. 6 shows the EPE hotspot reduction before and after RADAR correction using wire spreading as well as ripup and reroute (R&R) techniques.

(a) (b) (c)

Figure 6: (a) initial EPE; (b) EPE after wire spreading; (c) EPE after rip-up & reroute.

The details of EPE reduction and wirelength comparison are summarized in Table 1. It can be seen that the ripup and reroute (R&R) reduces EPE by up to 40% compared to wire spreading (SP). So wire spreading alone for post-routing lithography optimization is not effective. The intuition behind is that ripup and reroute surgically correct each EPE violating segment with accurate EPE contributing information from its neighbor, thus is more effective. On the other hand, for EPE based wire spreading, violating segment are untouched but we tend to uniformly spread its neighbors using more coarse objectives. Table 1 also shows that the wirelength increase is much less with R&R (RRwl) than wire spreading (SPwl). Our fast EPE simulation is very fast. For a block of 650K cells, in a 3mm x 3mm region, it only takes 135 seconds on an Intel Pentium machine running Linux (2.8 Ghz).

Table 1: Comparison 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 an effective RET aware detailed routing (RADAR) framework guided by fast lithography simulations. We introduce the concept of edge placement error map to guide RADAR. We propose two EPE guided detailed routing algorithms which are suitable for full-chip post layout optimization, one is wire spreading, and the other is blockage based ripup and reroute. Promising results are obtained using industrial circuits. In the future, we plan to incorporate other RET metrics and perform more correct-by-construction routing with high fidelity litho-metrics.

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, Hardy Leung from Magma, and Lida Huang from TI for their helpful discussions.

8. REFERENCES

[1] L. W. Liebmann, "Layout impact of resolution enhancement techniques: impediment or opportunity?", *Int. Symp. on Physical Design*, 2003.

[2] F. M. Schellenberg, "Resolution enhancement technology: the past, the present and extension for the future". *SPIE Microlithography Symposium*, 2004.

[3] P. Gupta and A. B. Kahng, "Manufacturing-aware physical design," *ICCAD*, pp. 681–687, 2003.

[4] H. K.-S. Leung, "Advanced routing in changing technology landscape," *Int. Symp. on Physical Design*, pp. 118–121, 2003.

[5] A. B. Kahng, "Research directions for coevolution of rules and routers," in Proc. *Int. Symp. on Physical Design*, pp. 122–125, 2003.

[6] L. Scheffer, "Physical CAD changes to incorporate design for lithography and manufacturability," *ASPDAC*, Jan. 2004.

[7] M. Lavin, F.-K. Luen, and G. Northrop, "Backend CAD flows for 'Restrictive Design Rules'", *ICCAD*, 2004.

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

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

[10] L.-D. Huang and D. F. Wong, "Optical proximity correction (OPC)-friendly maze routing," *DAC*, 2004.

[11] Y.C. Pati, A.A. Ghazanfarian, and R.F. Pease, "Exploiting structure in fast aerial image computation for IC patterns", *IEEE Trans. Semi. Mfg.*, Feb 1997.

[12] J. Stirniman and M. Rieger, "Fast proximity correction with zone sampling", in *Proc. SPIE Symposium on Microlithography*, vol. 2197, pp 294-301, 1994.

[13] N. B. Cobb, *Fast Optical and process Proximity Correction Algorithms for Integrated Circuit Manufacturing*, Ph.D. Thesis, UC Berkeley, 1998.

[14] Blast-Fusion, Magma Design Automation.