

Optical Lithography Simulation using Wavelet Transform

Rance Rodrigues, Aswin Sreedhar, Sandip Kundu
University of Massachusetts at Amherst
{rodrigues, asreedha, kundu}@ecs.umass.edu

Abstract—Optical Lithography is an indispensable step in the process flow of Design for Manufacturability (DFM). Optical lithography simulation is a compute intensive task and simulation performance, or lack thereof can be a determining factor in time to market. Thus, the efficiency of lithography simulation is of paramount importance. Coherent decomposition is a popular simulation technique for aerial imaging simulation. In this paper, we propose an approximate simulation technique based on the 2D wavelet transform and use a number of optimization methods to further improve polygon edge detection. Results show that the proposed method suffers from an average error of less than 5% when compared with the coherent decomposition method. The benefits of the proposed method are (i) >10X increase in performance and more importantly (ii) it allows very large circuits to be simulated while some commercial tools are severely capacity limited. Approximate simulation is quite attractive for layout optimization where it may be used in a loop and may even be acceptable for final layout verification.

I. INTRODUCTION

Optical Lithography is the process that involves the transfer of features from a mask onto a silicon wafer. During the process light is shone onto the mask pattern which makes an imprint on the resist that lies over the silicon wafer, on the image plane. The proper functioning of the circuit on the mask depends on the fidelity of pattern transfer onto the silicon wafer.

Technology scaling in semiconductor industry is mainly driven by corresponding improvements in optical lithography technology. Lately, improvements in optical technology have been difficult and slow. The transition to deep ultra-violet (DUV) light source (193nm) required changes in lens materials, mask blanks, light source and photoresist. Consequently, as the industry moves towards manufacturing end-of-the-roadmap CMOS devices, lithography is still based on 193nm light source to print critical dimensions of 65nm, 45nm and likely 32nm.

In sub-wavelength lithography, interference and diffraction can result in a corrupt pattern transfer, where proximity of other features around the feature being printed poses printability issues [1]. Optical lithography simulation is used to predict distortions such that they can be corrected during design. Unfortunately, litho simulation is notoriously slow. In this paper we explore an approximate simulation technique based on 2D wavelet transform, with the ultimate goal of improving performance and capacity of optical simulation. The method involves the use of a wavelet, the parameters of which are functions of the imaging system.

Studies on benchmark circuits show that this approximate model achieves results within 5% of commercial tools, while significantly increasing the capacity and performance of simulation. Commercial litho simulation tools have difficulty simulating layout of entire benchmark circuits, which the approximate technique handles with ease. This method can easily be extended to produce Optical Proximity Correction (OPC), Design Rule Check (DRC) and other algorithms that involve layout printability analysis.

II. BACKGROUND

Optical lithography simulation involves the process of obtaining the intensity of electromagnetic waves that have propagated from the light source, diffracted through the mask patterns and projected onto the wafer. Using electromagnetic equations directly in conjunction with finite difference or finite element method [2] is costly in terms of computation time. Originally proposed by Gamo [3], the decomposition of Hopkins partially coherent equations and edge lookup for convolution was used by Nick Cobb in his PhD thesis [4] to speed up aerial image simulation. Similar work was done by Mitra, Yu and Pan [5], in which edge lookup and decomposition of polygons in the mask into upper right corner rectangles was used to enable speedup in simulation.

In [6] and [7], the authors simplify the *Optical Coherent Approximation (OCA's)* [8], using the symmetric properties of the lithography imaging system. In [9], the authors show that the major source of error in generating the *Transmission Cross Coefficients (TCC)* is due to discontinuous boundaries of the TCC integrand. They report the use of improved numerical algorithms to dramatically improve speed and accuracy of simulation. In [10], Pati, Ghazanfarian and Pease use the incoherent sum of coherent imaging systems introduced by authors of [8], and further enhance simulation time by decomposing every mask into a set of commonly appearing shapes in IC's and then performing simulations based on pre computations of those shapes.

Hardware based acceleration techniques have also been investigated [11] and used in commercially available lithography simulation products showing a considerable speedup over methods using a single CPU for the simulation [12][13]. Our method is different from the TCC based methods in which we use wavelet transform to approximate the aerial image intensity and hence predict polygon edges post optical lithography.

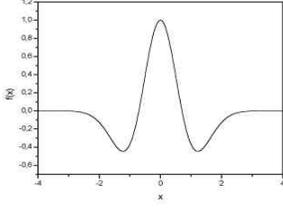


Fig. 1. Mexican Hat Wavelet.

III. ON WAVELET TRANSFORM

Wavelet Transform is a method similar to the Fourier transform in which a convolution is performed on a signal to convert it into something that is easier to analyze [14]. It is a well known technique in image processing to achieve compression of data, data frequency analysis etc. It involves the correlation of a signal (1D Transform) or an image (2D Transform) with a wavelet.

A wavelet is a wave like function that is time limited. The Mexican Hat wavelet is an example of such a wavelet as shown in Fig. 1. The equation for wavelet transform is given in (1). Here $x(t)$ is the mask function and ψ represents the wavelet used for the transformation.

$$T(a, b) = w(a) \int_{-\infty}^{\infty} x(t) \psi^* \left(\frac{t-b}{a} \right) dt \quad (1)$$

Here a and b are called the scaling and translational parameters of the transform respectively. They are used to calculate the transform at different locations of the signal b , and at different scales of wavelet a . $T(a, b)$ is the transform value or coefficient at a scale of a , location of b and $w(a)$ is called the weighing function used to ensure that the energy of the wavelet at every scale is constant [14].

The wavelet that has translation parameter $b = 0$ and scaling parameter $a = 1$ is called the mother wavelet. Daughter wavelets are obtained by assigning a and b values. Depending on the application, a signal or image can be broken up into wavelet coefficients using the mother and several daughter wavelets. In our method, we just use a single mother wavelet, and the wavelet used is the sinc^2 function, for reasons given in the following paragraphs.

Consider a setup as shown in Fig. 2. This is a simple single slit diffraction experiment. A lens of an arbitrary pupil shape is placed between mask plane and the image plane. When the rays pass through the lens, diffraction occurs and the rays then converge at the focal point of the lens. This phenomenon is called as Fraunhofer diffraction.

Due to constructive and destructive interference between the light waves, a sinc^2 pulse is obtained at the output instead of a rectangle. Similar wave formation is seen for masks that contain polygons. Since the aerial image of a single rectangular slit based on far-field Fraunhofer effect is a sinc^2 wave, we use the sinc^2 wavelet to approximate the aerial image at any given point. The sinc^2 function is then made a function of the parameters a and b , scale a chosen for the wavelet depending on the optical system parameters

and b depending on the location where the aerial image needs to be calculated.

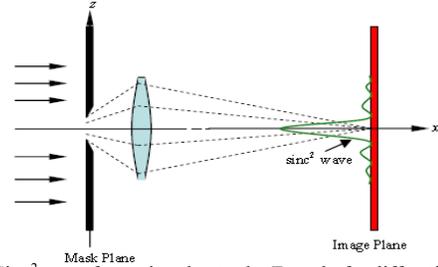


Fig. 2. Sinc^2 wave formation due to the Fraunhofer diffraction in a single slit experiment.

A. Aerial image intensity calculation

Aerial image intensity calculation is the first step in the process of lithography simulation. The accuracy of the overall simulation process depends on the accuracy of this step as the final edge locations of the polygons on the silicon wafer depends on the image intensity at the photoresist. The image intensity for our method is obtained by the wavelet transform (correlation) of the mask function $M(x, y)$ with the wavelet function $\psi(x, y)$, the 2D sinc^2 function in our case. The region of support for the wavelet is fixed to be equal to the optical diameter ($\sim 1\mu\text{m}$) as in Fig. 3. This parameter defines the region of influence around the current point under consideration, say (x, y) and the aerial image intensity calculated at this point is a function of the mask pattern within the optical diameter. The aerial image intensity calculation at any point b is obtained by placing the wavelet centered at that point, setting the scale of the wavelet and then a correlation of the wavelet with the portion of the mask within the optical diameter. The mask equation is given below in (2).

$$M(x, y) = \begin{cases} 0 : \text{feature present} \\ 1 : \text{feature not present} \end{cases} \quad (2)$$

For the purpose of our simulation, the wavelet is stored as a 2D mesh and the magnitude at any point is a function of the radius of the point from the point at which the center of the wavelet is placed. The mask is also stored in the same way as the wavelet.

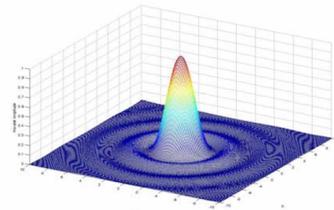


Fig. 3. The 2D sinc^2 function with support over a circular region.

The aerial image intensity value is then calculated as in equation (3) given below. Here $T(x, y, a)$ is the transform value at location (x, y) and scale of wavelet a . The wavelet is a function of the location and scale, while the mask function $M(x, y)$ is just a function of the location (x, y) . The choice of

wavelet scale is mentioned in the sections that follow.

$$T(x, y, a) = \sum_y \sum_x \psi(x, y, a) * M(x, y) \quad (3)$$

A pictorial description of the above equation is shown in Fig. 4. As the wavelet function is symmetrical, just one quadrant of the wavelet is calculated and used for all four quadrants of the image to reduce unwanted computation.

B. Edge detection

As mentioned earlier, the edges of the polygons printed on the silicon wafer do not match those on the mask due to a number of factors. The ultimate objective of lithography simulation is to detect accurately enough, the locations of these edges after lithography. The aerial image intensity, the key factor in edge detection, is the intensity of light just above the photoresist present over the silicon wafer. If the image intensity at a point on the resist exceeds a certain threshold, the resist is activated and undergoes chemical changes, while for points below the threshold effectively nothing happens. Depending on the chemical nature of the resist used, positive or negative, these are the regions that define the edges of polygons on the silicon wafer. The modeling of this phenomenon is done using the aerial image intensity. The constant threshold model [4] uses a single constant value, calculated from the normalized aerial image intensity profile as the *Edge Detection value* as defined in equation (4). This value is set once and then used to predict edges for the rest of the mask. We use this model which is also known as the 0.3 contour method. Consider the example in Fig 5, of an aerial image obtained for a single feature.

$$EdgeLocation = 0.3 * (Max - Min) + Min \quad (4)$$

The Max and Min values are calculated as the maximum and minimum values for the given aerial image. The edge detection value is then calculated as given as above in (4).

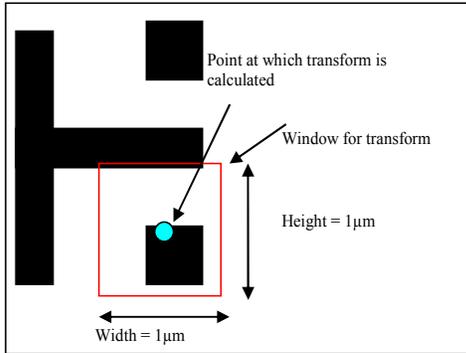


Fig. 4. Wavelet coefficient calculation at a point in the mask.

After we calculate the edge location value, we look for it in the aerial image and hence we get post-lithography polygon edge position using the 0.3 contour method. In Fig. 5. Min and Max are calculated as explained and the points marked by edge location in the figure are the locations at which the final edge placement of the polygons happens.

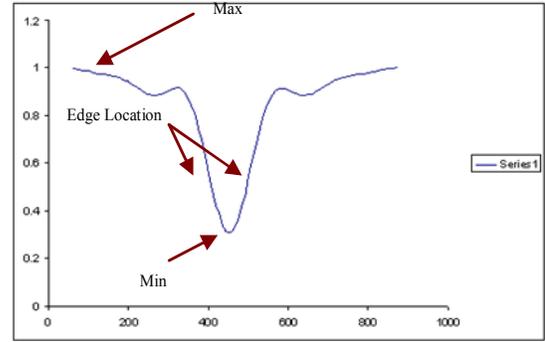


Fig. 5. Aerial image of a single feature used to detect feature edges.

IV. SOFTWARE SPEEDUP TECHNIQUES

Calculation of the transform coefficients is a time intensive process. In order to achieve a speed up, we use a number of software speed up techniques to simplify the calculation. We describe three methods where maximum benefits were observed.

Look for edge in this region

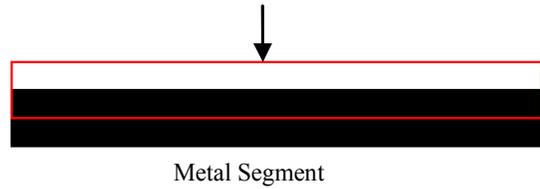


Fig. 6. Detect Post Litho Edges around Polygon Edges.

A. Reduce computation region

In calculating the final edge placement points for features on the mask, we need only a few simulation points as is seen in the 0.3 contour method for edge detection. It is observed that in most cases, the edge on the silicon wafer will lie more or less around the edge of the polygon in the mask. So the search begins around the edges of polygons in the mask for the edge detection. This is highlighted in Fig . 6. In order to further simplify the process of locating the feature edge, slope estimation method is used. We calculate the transform at 2 points and then using the slope of the line between them, the edge location is estimated, which in turn becomes the next simulation point, until the simulation converges. This technique produces a considerable speedup without any loss in accuracy.

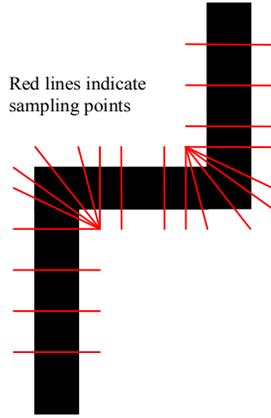


Fig. 7. Figure depicting potential high and low sampling regions.

B. Use variable sampling to reduce simulation points

In addition to limiting the region within which we search for edges, a sampling algorithm is used that determines the points along the edge of a polygon where wavelet simulation is performed to obtain the edges of the post-lithography contours. Such contour approximation technique further improves the simulation time. In regions where the aerial image is expected to be affected due to additional factors like close proximity of other features or along feature corners where problems like corner rounding and variations in line width and height may occur, a higher sampling rate is used for determining the edge of the contour. Fig. 7 and Fig. 8 show a couple of the many situations where variable sampling will be used along the edges of the features on the mask. At other regions where we do not have additional factors affecting the aerial image, we use a low sampling rate, as the edge location; post lithography at such locations on the mask will not vary much as we move from one point to the next. By using the sampling method, there is some loss of accuracy in the low sampling regions, but judicious selection of sample points reduces possibility of error in edge detection, while the simulation performance gain is significant.

C. Use spatial locality of transform coefficients

This is based on the fact that transform coefficient magnitude at a point is the function of the mask pattern in the window around the (x,y) location that we currently calculate the coefficient at. As we move from the current location to the next, unless we have a large change in x or y value, a portion of the mask pattern within the window at the next (x,y) location will be the same as before, indicating that the transform coefficient may not have changed much. Say that we are detecting an edge at a location (x_1, y_1) for a polygon edge extending in the x direction. If the edge for this point is found at a point (x_1, y_{edge}) , as we move along the x -axis to the next (x_2, y_1) location, the edge at that point will most likely be found at more or less location (x_2, y_{edge}) . We use this history information to speculate where the edge for

the current simulation point may lie.

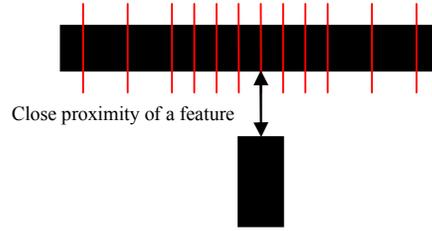


Fig. 8. Another potential high sampling region.

V. EXPERIMENTATION

The location of polygon edges after optical lithography depends on the focus of the optical system being used. We found in our experiments that the focus of the optical system being used is equivalent to the scale of the wavelet used in our system. In this section, we describe the method followed and the results in an effort to relate the scale of wavelet to the focus. For this paper we concentrate on using an optical system with focus set at 0.

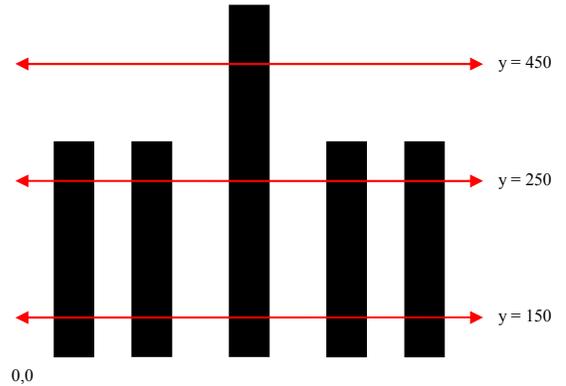


Fig. 9. Sample mask considered for relating the scale of the wavelet to the focus of the optical system. The red lines indicate the sections along which the aerial images obtained from the proposed model and the commercial software were compared.

Consider the sample mask in Fig. 9. To find the relation between the wavelet scale and the focus of light used, we ran wavelet simulations on a number of masks in a loop using different scales and each time compared our results with results from a commercially available lithography software tool. We show the analysis we performed for the mask in Fig. 9 and discuss the results. In Fig. 9, red lines indicate the sections at which we ran the simulation and compared results. We generated aerial images at those y locations and compared them with the results we had from the commercially available software. The Percentage Error metric used in the plots is defined as the sum of the difference of the aerial images obtained from our simulator and the commercial software at every point for the simulation section chosen. We found that on an average the wavelet scale parameter of 29 best produces a similar aerial image as that obtained from the commercial software at focus 0. The results are plotted in Fig. 10, where the percentage error is calculated for various scales at various

sections of the mask and Fig. 11, where the aerial images are shown obtained from both the simulators at $y = 250 \text{ nm}$ and scale chosen as 29.

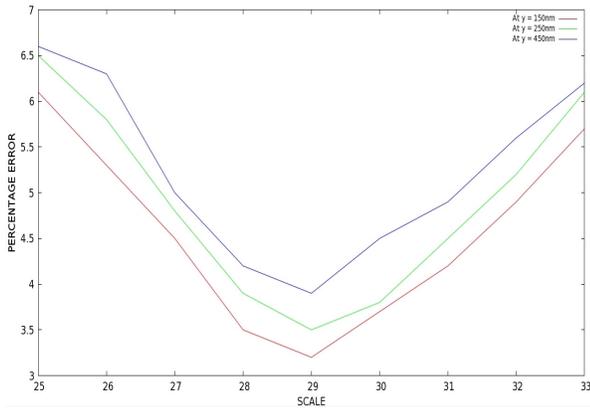


Fig. 10. Percentage Error Vs Scale for a sample mask at focus 0.

It can be seen from Fig. 11, the difference between the aerial images is such that final edge placement may result in more or less the same location as is predicted by the commercial software. This shows that scale of 29 for the wavelet as the best fit for a focus of 0 in the optical system.

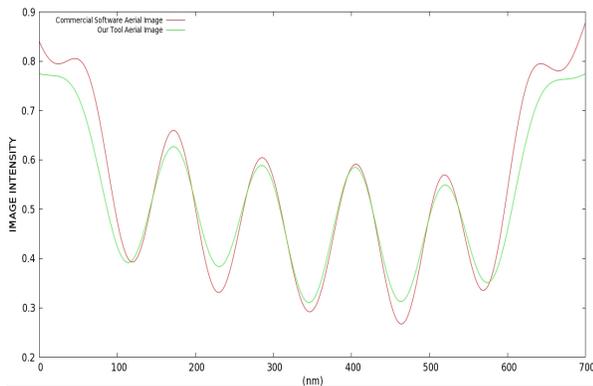


Fig. 11. Aerial image comparison at $y = 250 \text{ nm}$.

VI. RESULTS

The tool we developed to implement the proposed method was written in C++ and it spanned around 3000 lines. The tool was run using a HP Pavilion 6500 notebook with Fedora 10 operating system, Intel® Core™ 2 Duo T7500 @ 2.2GHz and 2GB of RAM. For small sample masks the results of final edge placement simulation are compared with the results obtained from the commercial software and our method. Fig. 12 shows the mask considered for comparison as well as the final edge placement using the two methods. The RMS error is defined as the RMS of the difference between the output from the commercial tool and our simulator. From the figure it can be seen that the proposed approximate method is close. Fig. 13 shows another sample mask we considered and the corresponding results. Once

again it can be seen that the final edge placement is highly close. The results for the 2 masks have been tabulated in table 1. This experiment has been repeated for many such masks but all the image results are not included here.

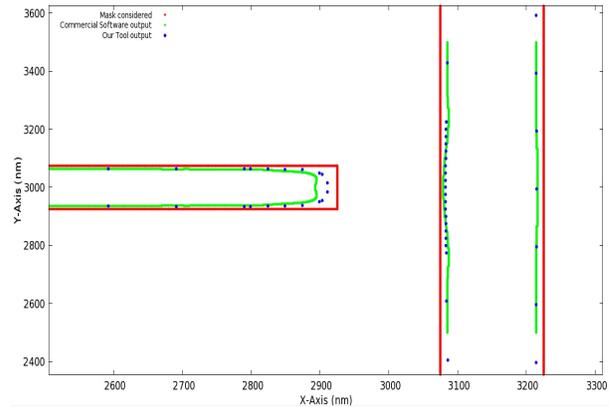


Fig. 12. Final Edge Placement comparison between the two methods using sample Mask 1.

Mask	Simulation Time (sec)		Speed up	% RMS Error
	Proposed Method	Commercial Software		
Mask 1	2	22	11.0	2.92
Mask 2	5	79	15.8	4.86

The real benefit of the proposed approach can be seen in larger circuits. For the purpose of our experiment, ISCAS-85 circuits were synthesized and mapped to a layout based on 45nm technology. We generated the GDSII layout files for the benchmarks using Cadence Encounter. For the larger circuits C17 and beyond, the accurate commercial simulator we have produced insufficient memory error while the proposed method was able to complete all of them as shown in table. 2.

VII. CONCLUSION

We investigated a wavelet based approximate optical lithography simulation method. The results suggest that the accuracy of our method, keeping in mind the various simplifications we use, on an average is within 5% while the average speedup we observed as compared to the commercial software for small masks is $> 10X$. Moreover, as the mask size increased the computation time using the commercial software increased quadratically, while it stayed mostly linear in the proposed method. A clear benefit of this approach is in improved capacity where the proposed method can run on larger circuits where commercial software fails. Based on partitioning techniques we have developed, the proposed method works on arbitrarily large GDSII layout files.

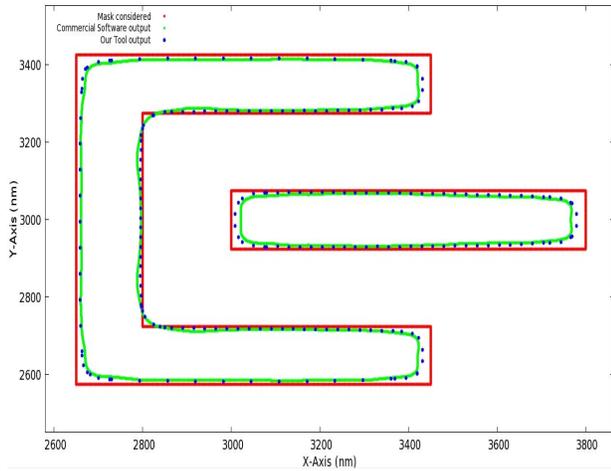


Fig. 13. Final Edge Placement comparison between the two methods using sample Mask 2.

VIII. FUTURE WORK

We look forward to speed up the system further by using multiple threads on multi core Processors. The investigation of the effect of change in focus, dose of light and light source on the wavelet is an interesting prospect and we plan to incorporate these parameters into the tool as the building process goes on. Process variation is another facet that we look forward to in the process of evaluating the performance of our simulator in various situations.

ACKNOWLEDGEMENTS

This work has been supported in part by a grant from the KLA-TENCOR Corporation and the Intel Corporation.

REFERENCES

- [1] A. Kahng, Y. Pati, "Subwavelength optical lithography: challenges and impact on physical design," Proceedings of ISPD, p.112, 1999.
- [2] M. S. Yeung. "Fast and rigorous three-dimensional mask diffraction simulation using battle-lemarie wavelet-based multiresolution time domain method". Optical Microlithography XVI, 5040(1):69–77, 2003.
- [3] H. Gamo. Progress in Optics, volume 3 ed. E. Wolf, chapter 3: Matrix Treatment of Partial Coherence, pages 187{326. North-Holland Publishing Company, 1964.
- [4] N. B. Cobb. Fast optical and process proximity correction algorithms for integrated circuit manufacturing. PhD thesis, UC Berkeley, 1998.
- [5] J. Mitra, P. Yu, and D. Z. Pan. "RADAR: RET-aware detailed routing using fast lithography simulations". In Proc. Design Automation Conference, pages 369–372, 2005.
- [6] P. Yu and D. Z. Pan, "A novel intensity based optical proximity correction algorithm with speedup in lithography simulation," in Proc. Int. Conf. on Computer Aided Design, 2007, pp. 854–859.
- [7] P. Yu, W. Qiu, and D. Z. Pan, "Fast Lithography Image Simulation By Exploiting Symmetries in Lithography Systems," IEEE Trans. on Semiconductor Manufacturing, vol. 21, no. 4, pp. 638–645, Nov. 2008.
- [8] Y. C. Pati and T. Kailath, "Phase-shifting masks for microlithography: Automated design and mask requirements," J. Opt. Soc. Amer. A, vol. 11, no. 9, pp. 2438–2453, 1994.
- [9] Yu, P.; Pan, D. Z. "ELIAS: An Accurate and Extensible Lithography Aerial Image Simulator With Improved Numerical Algorithms". Semiconductor Manufacturing, IEEE Transactions on Volume 22, Issue 2, May 2009 Page(s):276 - 289 Digital Object Identifier 10.1109/TSM.2009.2017652.
- [10] Y. C. Pati, A. A. Ghazanfarian, and R. F. Pease. "Exploiting structure in fast aerial image computation for integrated circuit patterns". IEEE Transactions On Semiconductor Manufacturing, 10 No. 1:62{74, Feb 1997.
- [11] Jason Cong , Yi Zou, "Lithographic aerial image simulation with FPGA-based hardware acceleration", Proceedings of the 16th international ACM/SIGDA symposium on Field programmable gate arrays, February 24-26, 2008, Monterey, California, USA [doi>10.1145/1344671.1344683].
- [12] Datasheet of calibre nmopc. Mentor Graphics.
- [13] Y. Cao, Y.-W. Lu, L. Chen, and J. Ye. Optimized Optical Microlithography XVIII, 5754(1):407–414, 2004.
- [14] Paul S Addison .The Illustrated Wavelet Transform Handbook

Table 2. Simulation run time for the ISCAS-85 benchmarks using the proposed method.

	C17	C432	C499	C880	C1355	C1908	C2670	C3540	C5315	C6288	C7552
1	12s	3m	2m34s	6m	3m43s	4m	6m	15m31s	15m45s	46m34s	19m16s
2	16s	13m2s	24m	26m21s	25m22s	28m27s	41m	1h33m58s	1h48m9s	3h40m23s	1h29m28s
3	14s	11m7s	21m6s	22m40s	22m	25m43s	42m6s	1h27m55s	1h53m3s	3h29m42s	1h39m11s
4	-	1m20s	4m33s	5m23s	5m	4m14s	10m11s	31m23s	31m18s	46m40s	24m29s
5	-	1s	40s	32s	37s	24s	1m40s	2m47s	10m23s	4m58s	4m40s
6	-	-	6s	1s	2s	-	21s	9s	2m6s	8s	1m47s
7	-	-	-	2s	2s	1s	4s	-	36s	12s	38s
8	-	-	-	4s	1s	-	-	-	8s	-	-

Note: The "-" in the table means that no such metal layer exists for the benchmark. Unit h-hours, m-minutes, s-seconds