A Kernel-Based DfM Model for Process from Layout to Wafer

Yiwei Yang*[a], Zheng Shi[a], Litian Sun[a], Ye Chen[b], Zhijuan Hu[a]
[a]Institute of VLSI Design, Zhejiang University, 38 Zheda Road, Hangzhou 310027, China
[b]Anchor Semiconductor Inc., 5403 Betsy Ross Drive, Santa Clara, CA 95054, USA

ABSTRACT

A layout design that passes the design rule check (DRC) may still have manufacturing problems today, especially around areas of critical patterns. Thus a design-for-manufacturability (DfM) model, which can simulate the process from designed layout to wafer and predict the final contours, is necessary. A new kind of DfM model called free-element-model (FEM) is proposed in this paper. The framework of FEM is borrowed from the forward process model, which is basically a set of convolution kernels in matrix form, yet the unknown variables are the kernel elements instead of process parameters. The modeling process is transformed into a non-linear optimization problem, with equality constraints which involve norm-2 regulation of kernels and inner production of any two kernels to keep the normalization and orthogonality of optimized kernels. Gradient-based method with *Lagrange* penalty function is explored to solve the optimization problem to minimize the difference between simulated contours and real contours. The dimension of kernels in FEM is determined by the cutoff frequency and the ambit. Since kernels are calculated by optimization method instead of decomposition of transmission cross coefficient (TCC), every element of kernels becomes a factor to describe the process. FEM is more flexible, and in it all effects that can be integrated into convolution kernels join naturally, such as the resist deviation and asymmetry of the process. No confidential process parameters, for example NA and defocus, appear in FEM explicitly, and thus the encapsulated FEM is suitable for IC manufacturers to publish. Moreover, enhancements and supplements to FEM are discussed in this paper, including the sufficiency of test patterns. In our experiments, DfM models for 2 process lines are generated based on test patterns, and the results show that the simulated shapes have an area error less than 2% compared to the real shapes of test patterns and an area error less than 3% compared to the shapes in typical blocks chosen from chip for verification purpose. The root mean square error of contour deviation between the 2 simulation results from FEM and conventional lithographic model is 10nm in a 65nm process.

Keywords: Design-for-Manufacturability (DfM), DfM Model, Manufacturability Verification, Optimization, Free-Element-Model (FEM)

1. INTRODUCTION

Design rules are the common language between designers and manufacturers; and as long as the layouts followed the rules, the manufacturability of layouts was guaranteed. However, as the IC manufacturing process keeps progressing, design rules become far-forth deficient to ensure the manufacturability of layouts today[1,2]. In fact, design rules are merely the foremost constraint to form design geometries and the layouts undergo further modifications after design tape-out. Such modifications typically include mask making, optical proximity correction (OPC) and scattering bar insertion (SBI). Those layout modification, simulation and verification tools (mainly CAD software), however, are not likely to be deployed in designer side and thus the designers know little information about details on manufacturability of their layouts. This reality now urges IC manufacturers to provide more information besides design rules to help the designers to find problems in early design stage. The additional information should be able to depict about the process form original design layout to final contour on wafer and it should also be well-encapsulated for safety reason.

This paper proposes a new design-for-manufacturability (DfM) model to describe the process from layout to contour. As an important supplement to design rules, this DfM model can simulate the final contour on wafer directly from original design layout without performing the layout modifications mentioned above. Neither the model presents any parameters

*yangyw@vlsi.zju.edu.cn

of the manufacture line, thus it is suitable for manufacturers to release to public. This DfM model can therefore be used in IC physical designing stage in order to reduce turn-around time due to manufacturability issues such as error-prone spots.

The rest of this paper is organized as follows. Section II describes and examines the overall framework of DfM modeling, including the problem formulation, problem solving and enhancements to the model. Experimental results are presented and discussed and the conclusion is drawn in Section III, followed by the discussion of future work and acknowledgements in Section IV and Section V.

2. DFM MODEL FOR PROCESS FROM LAYOUT TO CONTOUR

A layout design may be manufacturing-unfriendly even after it passes design rules check (DRC), thus the designers always want to know in design stage whether their layouts can be manufactured with high yield rate, and therefore to reduce the turn-around time. The DfM model proposed in this paper helps manufacturability verification in design stage in a degree as conventional DRC does, which will make the IC physical design flow more efficient and robust.

A mature process line has certain working conditions, whether the exposure dose, OPC recipe or implant energy should keep stable over specific environment. Though process variations widely exist[3,4], the procedure from design layout to final contour on wafer is approximated by a definite mapping function in this paper. As lithography being the dominating step in IC manufacturing, our DfM model adopts a similar structure of conventional lithographic simulation model, which from a mathematical view is a low-pass system with layout as the input and contour as the output. All process techniques designed to improve the performance of this low-pass system, for example, the sophisticated RET approaches, are considered describable within the overall mapping framework.

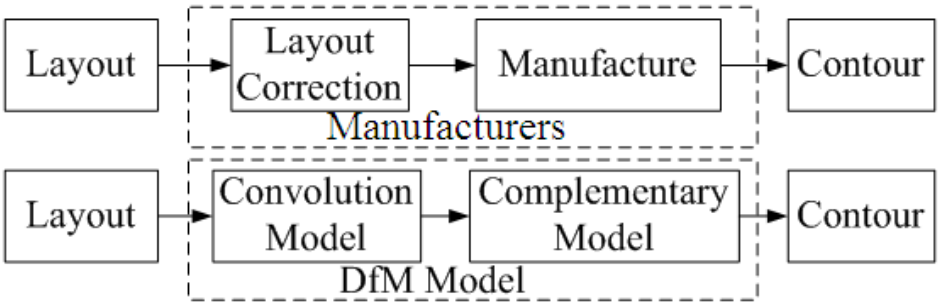


Figure 1. The schematic shows the procedure from layout to contour in IC manufacturers and our DfM model is to describe this whole procedure

Based on the above statements, a set of convolution kernels in matrix form, named *Convolution Model* part in Figure 1, are used to represent the primary relationship between design layout and final contour. The *Complementary Model* part, on the other hand, includes enhancements to further improve the DfM model's quality.

2.1 Problem Definition

We first define the symbols and variables. $M, Z \in D^{H\times L}$ denote the original layout image and its corresponding real wafer image in matrix form, whose element value is in set $D = \{0, 1\}$; H and L denote the size of these 2 matrices. $C \in V^{H\times L}$ denotes the simulated etched image in matrix form, with $C_{ij}\in[0,1]$ being the element value at row i and column j. $K_i \in R^{P\times P}$ denotes the i^{th} convolution kernel matrix with a dimension of $P\times P$ and $\lambda_i \in R$ denotes its corresponding weight. Etch-factor e and threshold t are constant factors used in a sigmoid function to transform an intensity image into an etched image.

According to our DfM model's framework, an intensity image matrix I ($I \in R^{H \times L}$) can be calculated by equation (1). This equation expresses a bi-linear convolution model and it agrees with the form of conventional optical model for a partially coherent system.

$$I = \sum_{i=1}^{N} \lambda_i (M \otimes K_i)^2 \tag{1}$$

where N denotes the number of used kernels, M denotes the mask matrix, K_i denotes the i^{th} kernel matrix and λ_i denotes its weight.

The complementary model, here the *constant threshold resist-like model* (CTR), can be approximated by a sigmoid function shown in (2), and by which we can keep the problem solving in continuous domain. In the same time, the element values of etched image matrix C distribute closely either to 0 or 1, which makes C the easier for comparing with the binary silicon image. Etch-factor e and threshold t both control the curve shape of this sigmoid function.

$$C = sigmoid(I) = 1/[1 + \exp(-e(I - t))] \tag{2}$$

where output C denotes the etched image matrix, input I denotes the intensity image matrix, e is the etch-factor and t is the threshold.

Following the preceding definitions, by substituting I into (2) the equation describing the entire process from layout to contour can now be rewritten into equation (3), which is the main DfM model proposed in this paper. The model parameters, or unknown variables, are essentially the convolution kernels K and their weights λ.

$$C(K_1, \lambda_1, ..., K_N, \lambda_N) = 1/(1 + \exp(-e(\sum_{i=1}^{N} \lambda_i (M \otimes K_i)^2 - t))) \tag{3}$$

To build this DfM model, a few original layout patterns and corresponding silicon contours, known as test pattern pairs, are needed. Then the DfM modeling can be rewritten into an optimization problem to minimize the difference between the silicon contours and simulated contours, as shown in (4).

$$\min cost(K_1, ..., K_N; \lambda_1, ..., \lambda_N) = \sum_{m=1}^{M} W_m \{\sum_i \sum_j (C_{m,ij} - Z_{m,ij})^2\} \tag{4}$$

where M denotes the number of test pattern pairs used to build model, W_m denotes the weight of the m^{th} pattern, $C_{m,ij}$ denotes the element at row i and column j of the simulated image matrix of the m^{th} pattern, $Z_{m,ij}$ denotes the element at row i and column j of the real silicon image matrix of the m^{th} pattern, $k_{n,ij}$ denotes the element at row i and column j of the matrix of the n^{th} kernel.

$$\sum_n \| K_n \| = \sum_n (\sum_i \sum_j K_{n,ij})^2 = 1, n = 1, ..., N \tag{5}$$

$$K_p \perp K_q = \sum_i \sum_j K_{p,ij} K_{q,ij} = 0, p \neq q, p = 1, ..., N, q = 1, ..., N \tag{6}$$

A few constraints are added to keep the kernels normalized and orthogonal, as shown in (5) and (6). Normalization means that through a transparent mask, the surface light intensity produced by the DfM model kernels should be 1. This constraint also ensures the threshold value t falling within the interval of (0, 1). Orthogonality means that the inner product of any two kernels should be 0, and this constraint helps to make the DfM model more compact.

$$\begin{aligned} &\min cost(K_1, ..., K_N; \lambda_1, ..., \lambda_N) = \sum_{m=1}^{M} W_m \{\sum_i \sum_j (C_{m,ij} - Z_{m,ij})^2\} \\ &s.t.\ \ nomalization\_constraint: \\ &\qquad \sum_n \lambda_i * \| K_n \| = \sum_n [\lambda_i * (\sum_i \sum_j K_{n,ij})^2] = 1, n = 1, ..., N \\ &\quad orthogonality\_constraint: \\ &\qquad K_p \perp K_q = \sum_i \sum_j K_{p,ij} K_{q,ij} = 0, p \neq q, p = 1, ..., N, q = 1, ..., N \end{aligned} \tag{7}$$

Up to now, the DfM modeling problem has been rewritten into an optimization problem of objective function *cost* with constraints, as shown in (7). The target of the DfM model optimization is generally the weight and elements of every kernel matrix. The input data needed for the DfM modeling are sufficient test pattern pairs and parameters in the complementary model. Here the test pattern pairs are original design layouts and corresponding silicon contours, which are to provide a good coverage of commonly seen shapes and typical cell / interconnect geometries.

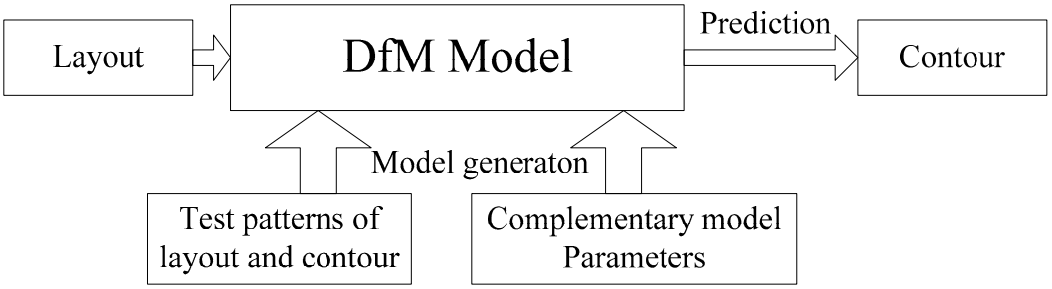


Figure 2. Schematic of DfM model generation and prediction.

2.2 Problem Solving

Since (7) is a non-linear optimization problem with equality constraints, gradient-based method with *Lagrange* penalty function can be employed to find optimal solutions, through which (7) is converted into an unconstrained non-linear optimization problem shown in (8). The objective function *new_cost* in (8) consists of three parts, namely the *cost* function in (7), the *normalization penalty term* and the *orthogonal penalty term*.

$$\begin{aligned} &\min new\_cost(K_1,...,K_N;\lambda_1,...,\lambda_N); \\ &new\_cost(K_1,...,K_N;\lambda_1,...,\lambda_N) \\ &= cost + normalization\_penalty + orthogonal\_penalty \\ &= \sum_{m=1}^{M} W_m \{\sum_i \sum_j (C_{m,ij} - Z_{m,ij})^2\} + [(\sum_n \lambda_i * \| K_n \|) - 1]^2 + \sum_{p=1}^{N} \sum_{q=p+1}^{N} (\sum_{ij} K_{p,ij} K_{q,ij})^2 \end{aligned} \tag{8}$$

Solving unconstrained non-linear optimization problem (8) needs to calculate the first order derivatives matrix of *new_cost* with respect to the i^{th} kernel matrix K_i and its weight λ_i. The iterative process is shown in (9).

$$\begin{cases} K_1^{n+1} = K_1^n - s * d_{K_1}^n \\ \lambda_1^{n+1} = \lambda_1^n - s * d_{\lambda_1}^n \\ ... \\ K_N^{n+1} = K_N^n - s * d_{K_N}^n \\ \lambda_N^{n+1} = \lambda_N^n - s * d_{\lambda_N}^n \end{cases} \tag{9}$$

where K_i^n denotes the i^{th} kernel matrix in the n^{th} iteration. s is step size. $d_{ki}^n \in R^{P \times P}$ denotes the first order derivative matrix of objective function *new_cost* with respect to kernel matrix K_i; likewise, $d_{\lambda i}^n \in R$ denotes the first order derivative of objective function *new_cost* with respect to kernel weight λ_i.

d_{ki}^n and $d_{\lambda i}^n$ can be calculated using the following equations shown in (10), similarly to (7) the derivatives in (10) all consist of three parts, namely the derivative of *cost*, of *normalization_penalty* and of *orthogonal_penalty*.

$$\begin{cases} d_{K_1}^n = \nabla_{K_1} cost + \nabla_{K_1} normalization\_penalty + \nabla_{K_1} orthogonal\_penalty \\ d_{\lambda_1}^n = \nabla_{\lambda_1} cost + \nabla_{\lambda_1} normalization\_penalty + \nabla_{\lambda_1} orthogonal\_penalty \\ ... \\ d_{K_N}^n = \nabla_{K_N} cost + \nabla_{K_N} normalization\_penalty + \nabla_{K_N} orthogonal\_penalty \\ d_{\lambda_N}^n = \nabla_{\lambda_N} cost + \nabla_{\lambda_N} normalization\_penalty + \nabla_{\lambda_N} orthogonal\_penalty \end{cases} \tag{10}$$

where

$$\begin{cases} \nabla_{K_i} cost = 4 * \lambda_i * e * \sum_{m=1}^{M} \{[W_m * (C_m - Z_m) \odot C_m \odot (1 - C_m) \odot (K_i \otimes M_m)] \otimes rot180(M_m)\} \\ \nabla_{K_i} normalization\_penalty = 4 * \lambda_i * \{\sum_{n=1}^{N} [\lambda_i * (\sum_{i=1}^{P}\sum_{j=1}^{P} K_{n,ij})^2] - 1\} * (\sum_{i=1}^{P}\sum_{j=1}^{P} K_{n,ij}) * one(P,P) \\ \nabla_{K_i} orthogonal\_penalty = 2 * \sum_{j,j\neq i} [\sum_{r=1}^{P}\sum_{s=1}^{P} K_{i,rs} K_{j,rs}] \odot K_j \\ \nabla_{\lambda_i} cost = 2e * \sum_{m=1}^{M} \{W_m * [\sum_i \sum_j (C_{m,ij} - Z_{m,ij}) * C_{m,ij} * (1 - C_{m,ij}) * (K_i \otimes M_m)^2]\} \\ \nabla_{\lambda_i} normalization\_penalty = 2 * \{\sum_{n=1}^{N} [\lambda_i * (\sum_{i=1}^{P}\sum_{j=1}^{P} K_{n,ij})^2] - 1\} * (\sum_{i=1}^{P}\sum_{j=1}^{P} K_{n,ij})^2 \\ \nabla_{\lambda_i} orthogonal\_penalty = 0 \end{cases} \quad (11)$$

where rot180(.) means to rotate the input matrix 180 degree, one(P,P) denotes a $P \times P$ size matrix with all its elements set to 1, * denotes the multiplying operator, $\odot$ denotes the operator of multiplying two matrices element-by-element, and $\otimes$ denotes the operator of convolution.

2.3 Problem Discussion and Model Enhancements

From the above discussions, we show that our DfM modeling problem is rewritten into a constrained non-linear optimization problem and a DfM model can be generated by solving this problem. The DfM model's form is similar to that of conventional lithographic simulation model; nevertheless the two are different in many aspects. In conventional lithographic simulation model, the model kernels are decomposed from transmission cross coefficient (TCC) and tuning model is to adjust the optical parameters such as NA and defocus. While in our DfM model (termed as free-element-model *FEM* in the paper), kernel elements are directly calculated, and every element is considered as a free variable to describe the manufacturing process; and therefore our DfM model has more elasticity. Moreover, all other effects which can be integrated into convolution kernel form could be included naturally in this DfM model.

On the other hand, since the modeling of FEM becomes an optimization problem, other mathematic methods can also be applied, for example *simulated annealing* (SA) method. After considering the problem size and accuracy requirements for the modeling, we find the gradient-based method easy and practical in terms of the optimizing time and parameter (e.g., the temperature and loop control in SA) setting. The optimization time and accuracy depend on the following factors in FEM modeling – the choice of constants and initial values of functioning parameters in the gradient-based method, sufficiency and coverage of test patterns, discretization method of layout and silicon shape, the choice of kernel numbers and the choice of the complementary model. To name a few in details, the constants and initial values of parameters that need to be selected are for etch-factor e, threshold t, step size s, the number of model kernels and elements of each kernel matrix.

Given the fact that FEM is generated completely based on the test patterns including design layout and silicon shapes, the sufficiency of the input modeling data is very important[5] – only pitch and space patterns are not enough, more complicated 2D patterns are needed to cover various geometrical possibilities. 1D gauge data (mainly the CD data) could be added to the objective function in (7) with respective weights representing their importance upon user's requirements. In our practice, input patterns include layout portions from standard logic cells, SRAM bit cell and lithographic model calibration sets.

Complementary model is critical for further improving FEM quality. In FEM generation flow, *constant threshold resist-like model* (CTR) is chosen as the default complementary model depicting resist behavior. CTR is straightforward yet effective in practice. To further reduce the fitting error of FEM, other complementary models can be explored and used.

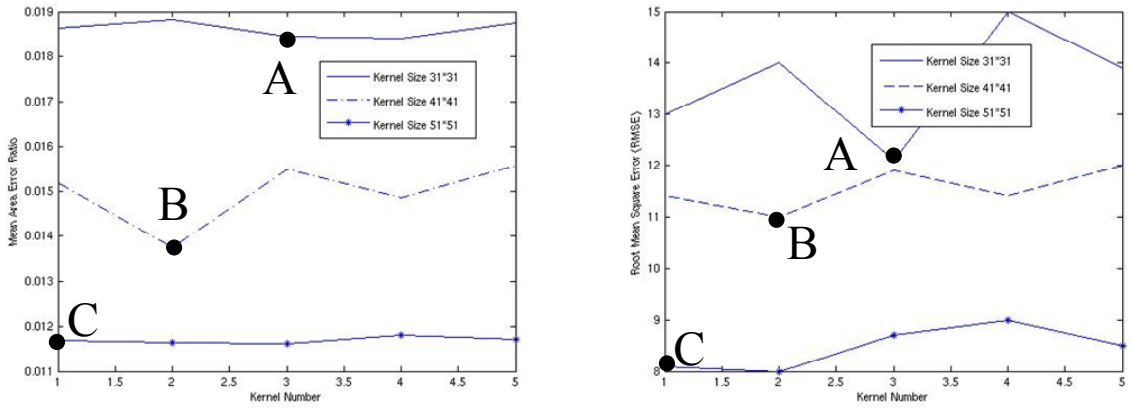
3. EXPERIMENTS AND DISCUSSIONS

We began our experiments on modeling and verification of FEM method firstly on a 160nm process using 248nm scanner. After that, we extended the experiments to a 65nm process using 193nm scanner.

In experimenting on the 160nm process and grid size being set to 10nm, 3 FEM model groups are generated with kernel size of 31×31, 41×41 and 51×51, respectively; within each group, there are 5 FEM models with kernel number varying from 1 to 5. The test patterns for FEM generation are from random logic and several types of arrays. The layout portions used in prediction and verification are different from those used for model generation.

We have measured the area error ratio and mean point placement error (PPE, for contour point placement error and measured evenly along the original polygon edges), between the silicon contour and the predicted contour based on generated FEMs. The results are shown in Figure 3, in which the horizontal axis is the number of kernels, while the vertical axis is the area error ratio in (a) and root mean square error (RMSE) of PPE in (b). In Figure 3(a) as well as in 3(b), the upper data line describes results for FEM with kernels size of 31×31; the middle data line is for kernel size of 41×41; and the data line on the bottom is for kernel size of 51×51. A conclusion could be drawn from these figures that the area error ratio and RMSE of PPE between the silicon contour and the simulated contour both decrease with kernel size, but do not change much with the number of kernels, i.e., the horizontal axis.

In Figure 3(a) and 3(b), data point A with kernel size of 31×31 and 3 kernels, data point B with kernel size of 41×41 and 2 kernels, and data point C with kernel size of 51×51 and only one kernel are chosen as the best among the 3 FEM model groups of different kernel size. Table 1 shows the values of area error ratio, line-end shortening and PPE measured on none line-end places using these 3 individual FEMs; the error data on model generation layout and model verification layout are also compared.



(a) (b)

Figure 3. Area error ratio (a) and mean PPE (b) vary with kernel size and kernel number in similar tendency.

Table 1. Experiment results of FEM generation and prediction in a 160nm process. μ is the mean value and σ is the root mean square error (RMSE)

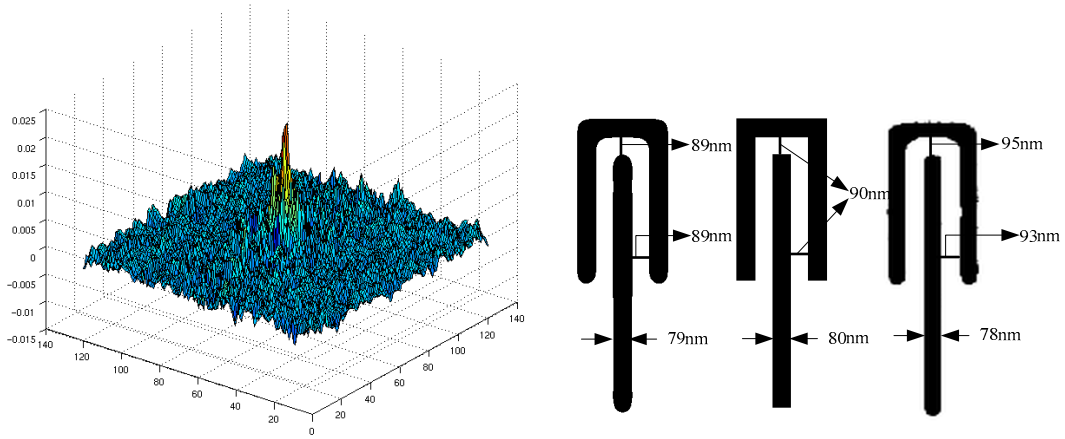
Best FEM from 3 size groups	**Model Generation**			**Contour Prediction**		
	ΔArea (%)	Line-end (μ, σ) (nm)	PPE (μ, σ) (nm)	ΔArea (%)	Line-end (μ, σ) (nm)	PPE (μ, σ) (nm)
3 kernels with size of	1.84	(7, 21.3)	(5.4, 12.1)	2.02	(7.3, 25.3)	(4, 16)

31×31						
2 kernels with size of 41×41	1.37	(5.8, 17.1)	(3, 11)	1.40	(6.2, 17)	(3.3, 12)
1 kernels with size of 51×51	1.16	(3.1, 13)	(3.1, 8.1)	1.19	(6.1, 14)	(2.7, 10.6)

In the second experimental process of 65nm technology, we use 11 test patterns for model generation and 22 layout portions for model verification. Kernel grid is set as 10nm and kernel size is set to 121×121 in this experiment. In Table 2, we still focus on the line-end and contour edge deviation to test the differences between simulated contour and target contour. The overall RMSE of PPE is 10nm in prediction while the mean value of PPE is 1.5nm.

Table 2. Experiment results of FEM generation and prediction in 65nm. The *Test Pattern* column lists the test pattern used for model generation. The *Layout Portion* column lists the layout portion used for model verification. μ is the mean value, σ is the root mean square error (RMSE).

Test Pattern	**Model Generation**			**Layout Portion**	**Contour Prediction**		
	ΔArea (%)	Line-end (μ, σ) (nm)	PPE (μ, σ) (nm)		ΔArea (%)	Line-end (μ, σ) (nm)	PPE (μ, σ) (nm)
1	1.16	(-1.2, 9)	(1.1, 7.7)	1	1.59	(-1, 13)	(0.8, 7.1)
2	0.94	(-2.0, 11)	(-0.8, 5.5)	2	1.76	(2.1, 7.1)	(1.2, 12.1)
…	…	…	…	…	…	…	…
11	0.54	(-4.8, 10.2)	(0.5, 6.4)	22	0.96	(1.3, 11.7)	(-1, 9.6)



(a) (b)

Figure 4. (a) is the generated kernel in the experiments of 65nm process. In (b), the middle one is original layout, the left one is the target contour and the right one is the simulated contour.

4. FUTURE WORKS

To obtain more accurate results for FEM prediction, smaller grid size can be used with the cost of much more optimizing time. Other complementary models, such as variable bias-like model (VBM), would be explored to further improve FEM

quality. To accelerate the modeling speed, parallel computing can also be adopted. In this paper, the approximating function (known as convolution model) is fixed, and the optimization process is to calculate unknown variables in this function. For higher level analysis of the whole system from layout to contour, the entire function space should be explored to find the most effective approximating functions; in other words, the approximating function itself becomes an unknown variable.

5. ACKNOWLEDGEMENTS

The authors would thank Dr. Xuelong Shi in SMIC and Prof. David Pan in UT Austin for precious suggestions and discussions.

REFERENCES

1. Clair Webb, “Intel design for manufacturing and evolution of design rules,” Proc. SPIE 6925, 692503 (2008)
2. Vito Dai, Luigi Capodieci, Jie Yang, and Norma Rodriguez, “Developing DRC plus rules through 2D pattern extraction and clustering techniques,” Proc. SPIE 7275, 727517 (2009)
3. Scott Mansfield , Ioana Graur, Geng Han, Jason Meiring, Lars Liebmann and Dureseti Chidambarrao, “Lithography simulation in DfM – achievable accuracy versus requirements,” Proc. SPIE 6521, 652106 (2007)
4. Scott Mansfield1 and Geng Han, Mohamed Al-Imam and Rami Fathy, “Through-process modeling in a DfM environment,” Proc. SPIE 6156, 615603 (2006)
5. Xuelong Shi, J. Fung Chen, Doug Van Den Broeke, Stephen Hsu, and Michael Hsu, “Quantification of two-dimensional structures generalized for OPC model verification,” Proc. SPIE 6518, 65180A (2007)