

Machine Learning-Guided Etch Proximity Correction

Seongbo Shim and Youngsoo Shin, *Senior Member, IEEE*

Abstract— Rule- and model-based methods of etch proximity correction (EPC) are widely used, but they are insufficiently accurate for technologies below 20nm. Simple rules are no longer adequate for the complicated patterns in layouts; and models based on a few empirically determined parameters cannot reflect etching phenomena physically. We introduce machine learning to EPC: each segment of interest, together with its surroundings, is characterized by geometric and optical parameters, which are then submitted to an artificial neural network that predicts the etch bias. We have implemented this new approach to EPC using a commercial OPC tool, and applied it to a DRAM gate layer in 20-nm technology, achieving predictions that are 34% more accurate than model-based EPC.

Index Terms— Etch proximity correction (EPC), machine learning, artificial neural network

I. INTRODUCTION

THE ETCHING process is a key step in semiconductor manufacturing: a resist image is created on a wafer by optical lithography (Fig. 1(a)); and, during the subsequent etching process, particles (e.g. ions and radicals) collide with the surface of the resist and the substrate (Fig. 1(b)). The problems that can potentially occur during etching process include etch proximity effects, which can produce an over-etched substrate (negative etch bias) due to lateral erosion of the photo resist, or an under-etched substrate (positive etch bias) due to excessive deposition (Fig. 1(b)). The extent of over- or under-etching is called the etch bias. To obtain a pattern of correct size on a wafer, a design layout has to be modified, using a process called etch proximity correction (EPC), which compensates for etch bias before the design layout is submitted to optical proximity correction (OPC); these processes are shown in Fig. 2.

A. Rule- and Model-Based EPC

Etch bias is affected by the quantity of etching particles and their angle and direction when the particles collide with the surface of the substrate [2]. A rigorous simulation of the etch proximity effect over an entire chip is not feasible due to the amount of computation required, so EPC has to be based on

Manuscript received August 10, 2016. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2015R1A2A2A01008037). Preliminary version of this paper [1] was presented at SPIE Advanced Lithography, San Jose, CA, February 21-25, 2016

The authors are with the Department of Electrical Engineering, KAIST, Daejeon 34101, Korea (e-mail: sbshim@kaist.ac.kr; youngsoo@ee.kaist.ac.kr). S. Shim is also with Samsung Electronics, Hwasung 18448, Korea.

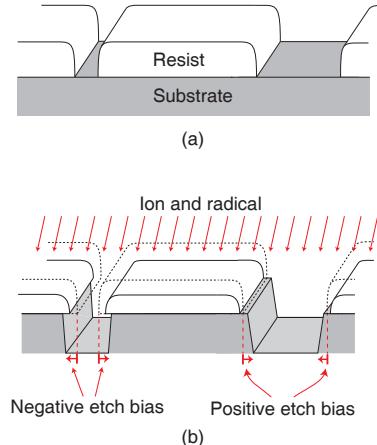


Fig. 1. (a) Resist image with small and large spaces and (b) corresponding etch result with negative and positive etch biases.

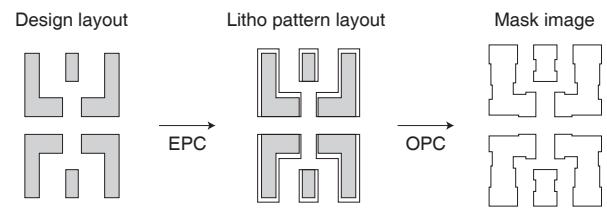


Fig. 2. EPC and OPC.

heuristics, which may be empirical rules (rule-based EPC) or simplified models (model-based EPC).

In rule-based EPC (RB-EPC), an established set of synthetic patterns, such as the dense line and space (DLS), isolated line (Iso), and line-end to line-end (E2E) patterns shown in Fig. 3(a), are created on a wafer; the etch bias is then measured at the center of each pattern; and the results are tabulated against the parameters of the patterns, e.g. their width and space. RB-EPC has been used for circuits with a small number of technology nodes [3], [4], but its rules are too simplistic to correct the range of pattern shapes that occurs in a real layout. It is estimated [4] that on-chip variation (OCV) in 10nm technology remains 28% out of tolerance after RB-EPC has been applied.

In model-based EPC (MB-EPC), etch bias is modeled as a function of a few empirical parameters:

$$\text{Etch bias} = C_0 + C_1 \mathbf{Den} + C_2 \mathbf{Vis} + C_3 \mathbf{Blo} + C_4 \mathbf{Den}^2 + \dots, \quad (1)$$

where **Den** is the density of the layout within a density kernel

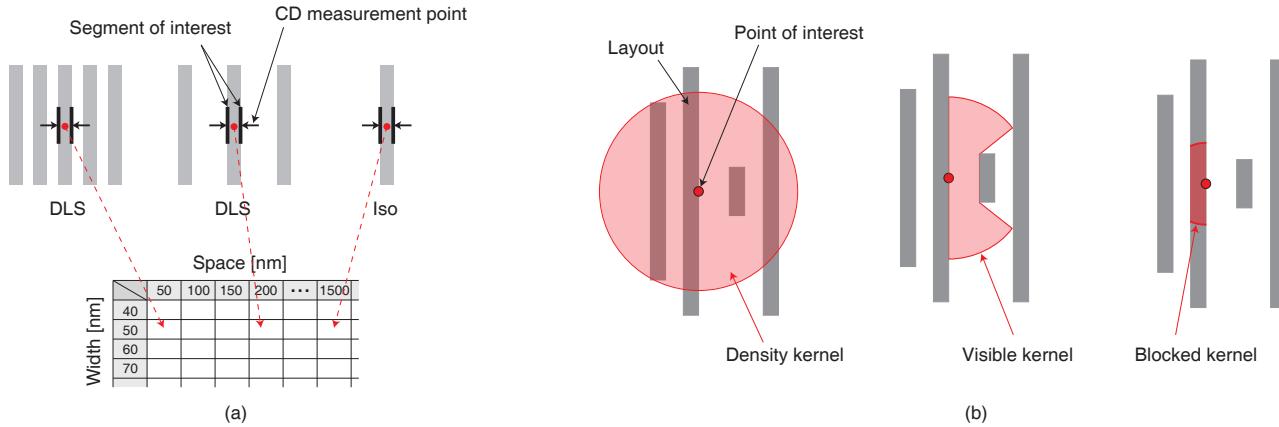


Fig. 3. (a) Synthetic patterns and etch bias rule table used in RB-EPC, and (b) kernels for MB-EPC.

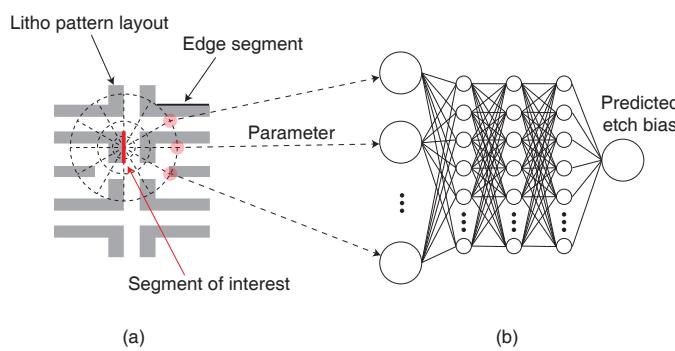


Fig. 4. The proposed etch bias model: (a) each segment in a layout is parameterized, and (b) an ANN is constructed to predict etch bias from input parameters.

region; **Vis** is the area of the open space that is not hidden by the edges that are neighbors of a point of interest (i.e. space beyond the nearest edge is ignored); **Blo** is the area of the nearest polygon that overlaps with the blocked kernel, as shown in Fig. 3(b) [5]. The coefficients C_i and the values of the number of terms in this function are determined empirically, and the radii of the kernels and the values of the coefficients C_i are adjusted through regression. MB-EPC can deal with a greater range of patterns than RB-EPC, but it still fails to achieve a satisfactory OCV. It is estimated that OCV in 20nm DRAM devices is still 15% of gate size after MB-EPC has been applied [5].

B. ML-EPC

We propose the use of machine learning techniques to achieve more accurate prediction of etch bias. Our approach is illustrated in Fig. 4. A litho pattern layout is divided into edge segments. A number of geometric and optical parameters are then extracted from each segment by examining its surroundings (Fig. 4(a)). These parameters are then inputted to an artificial neural network (ANN), which outputs the predicted etch bias of the segment (Fig. 4(b)). This ANN is trained on a number of segments that we prepared so that their etch biases

can be predicted as correctly as possible. The trained ANN is effectively an etch bias model, but its accuracy and efficiency depend on the parameters that we choose, the set of training segments used for ANN construction, and the complexity of the ANN.

Our main contributions can be summarized as follows:

- A choice of parameters, such as local pattern densities and optical kernel signals, which represent the physical phenomena of etching accurately, while being relatively simple to compute.
- A systematic method of segment preparation for comprehensive coverage of the complicated patterns that occur in a real layout.
- Optimization of an ANN through k -fold cross-validation to avoid over-fitting.

The remainder of this paper is organized as follows. In Section II, we describe the four steps in ML-EPC. In Section III, we show how ML-EPC performs on some test layouts of DRAM gate layers in 20nm technology. In Section IV, we draw some conclusions.

II. ML-EPC STEPS

A. Preparation of Training Segments

We used two types of pattern to prepare training segments: standard synthetic patterns and more complicated patterns extracted from layouts of real circuits. Standard synthetic patterns (see Fig. 3(a)) can be analyzed to see how etch bias changes in response to changes in geometry, e.g. a space of DLS patterns. For example, if etch bias appears to change abruptly in response to a gradual change in geometry, we can infer that there are errors in the measured data, and that we need to reexamine the measurement process.

The etch bias of a synthetic pattern is determined as follows: We measure a critical dimension (CD), which is the width of a pattern (or space) between two segments placed face-to-face, as shown in Fig. 3(a), after lithography (CD_{litho}) and then again after the subsequent etching (CD_{etch}). Since all the synthetic patterns are symmetric, the two segments can be assumed to have the same etch bias, and thus the etch bias of one segment can be expressed as $(CD_{etch} - CD_{litho})/2$.

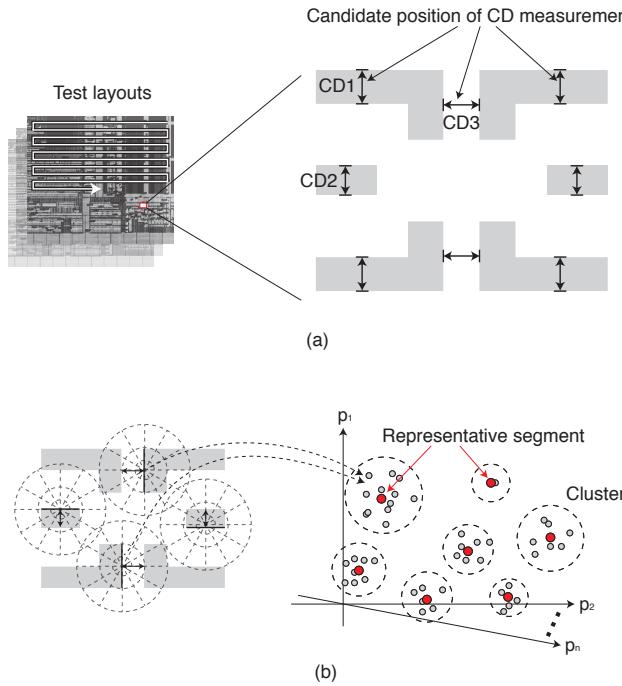


Fig. 5. Preparations of complicated training segments: (a) segments extracted from test layout and (b) representative segments after classifying the segments.

More complicated patterns are obtained from circuits as follows: Test layouts are prepared by shrinking layouts produced with earlier technologies, or synthesizing designs from a preliminary version of the new technology library. Pairs of segment between which to measure CD (see candidate position of CD measurement) are then identified on parallel edges, as shown in Fig. 5(a). Each candidate position is then examined to determine whether the two associated segments and their surroundings are symmetric; if they are symmetric (e.g. CD2 and CD3 in Fig. 5(a)), then they should acquire the same etch bias, which is expressed as $(CD_{\text{etch}} - CD_{\text{litho}})/2$; otherwise (e.g. CD1 in Fig. 5(a)), they are likely to have different etch biases, and neither bias can be readily identified: such points are rejected. Identical, or very similar, points are weeded out through a classification process. The segments corresponding to the remaining points are parameterized and mapped to points in an n -dimensional parameter space, as shown in Fig. 5(b). Some close points are then grouped through modified K -mean clustering [6], [7], and representative points are selected from each cluster: the corresponding segments are used to train our etch bias model.

B. Choosing Parameters to Characterize a Segment

Two types of parameter are used to characterize a segment and its surroundings. The first parameter is a local pattern density, which is measured at points sampled around the segment of interest. Fig. 6 shows how we draw lines passing through the center of the segment, and place a series of touching circles along each line. The radii r of these circles, which are the regions in which pattern density will be measured, increase by a user-defined ratio as circles are placed further from the

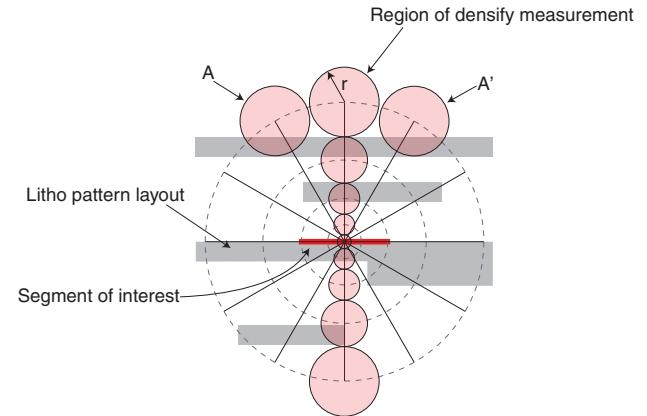


Fig. 6. Parameterization of a segment using local pattern densities.

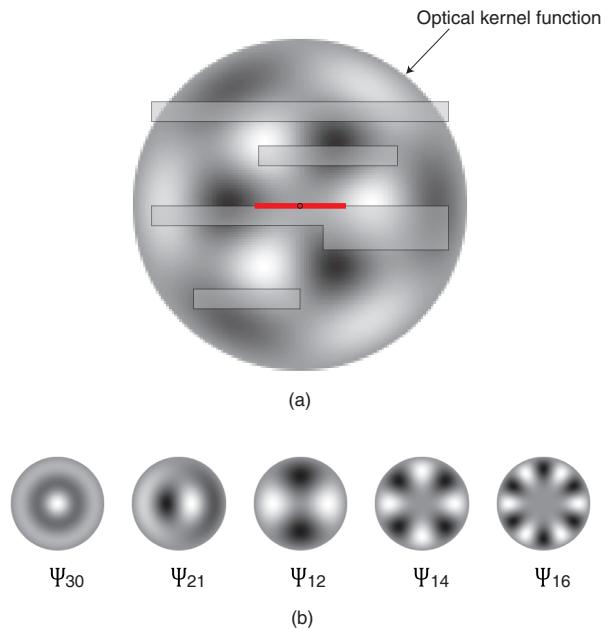


Fig. 7. (a) Parameterization of a segment using optical kernel signal and (b) some optical kernels.

segment. The total number of circles determines the number of parameters. A local pattern density is associated with the quantity of etching particles, and the position at which local pattern density is measured is associated with incident angle and direction of the particles. Because etching particles with the same incident angle and the symmetric directions equally affect etch bias, density values at symmetric locations (e.g. A and A' in Fig. 6) can be summed and regarded as a single parameter, which reduces the number of parameters, making it faster to train the ANN.

To take account of sidewall angle of resist, we choose an optical kernel signal ϕ_{nm} as the second parameter type, which is given by

$$\phi_{nm} = \sum_{\forall(x,y)} L(x,y)\Psi_{nm}(x,y), \quad (2)$$

where L is a pixelated image of the layout, in which the value of a pixel is 1 if it is within a layout polygon, and 0 otherwise. Ψ_{nm} is an optical kernel function, centered on an interesting

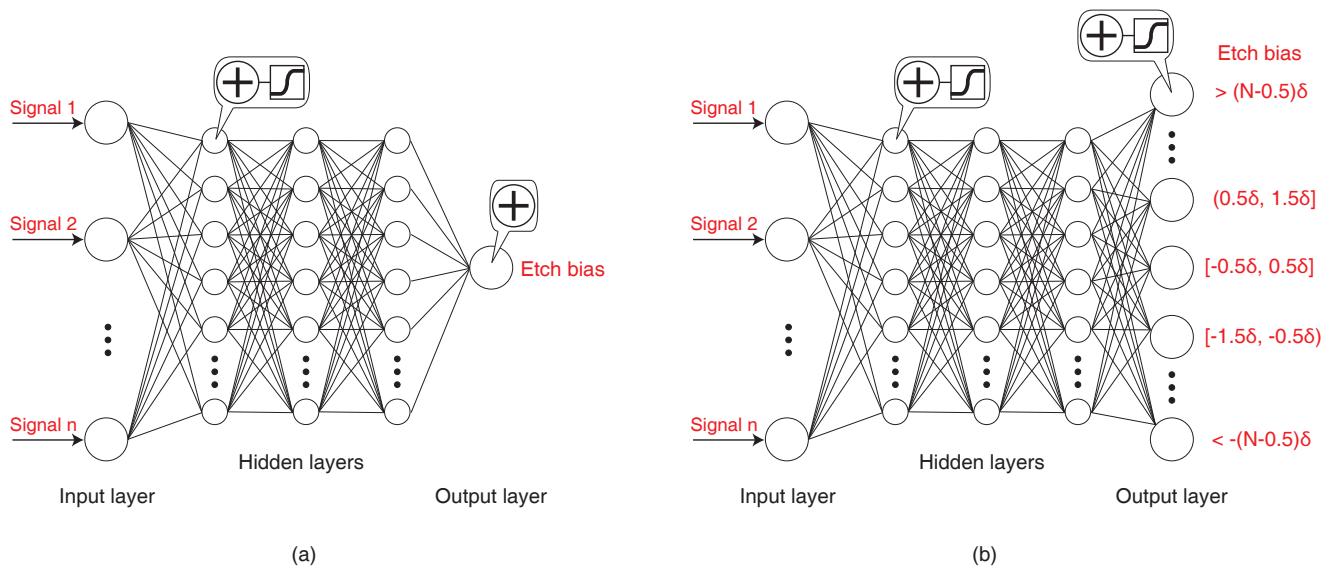


Fig. 8. ANN for (a) regression and (b) classification.

segment as shown Fig. 7(a), and expressed as follows:

$$\Psi_{nm}(r, \varphi) = J_n(r)\cos(m\varphi), \quad (3)$$

where J_n is n^{th} Bessel function, which is a radial component of the optical kernel function; and the value of n is the number of critical points in the radial direction. The angular component of the optical kernel function is represented by the cosine function¹, where φ is the angle and m is the number of cycles through this angle. The optical kernel function becomes more complicated as n and m increase, as shown in Fig. 7(b). Optical kernel signals model constructive or destructive interference of light at the center of the segment, and thus are associated with the vertical shape of the resist, which affects the etch bias [8]. Since each function makes a separate contribution to the resulting light intensity, we choose those functions which make the largest contribution.

C. ANN Construction

An artificial neural network (ANN) consists of layers of nodes connected by edges as shown in Fig. 8. The input layer of our ANN has n nodes, each of which receives a parameter from an interesting segment. Each input node propagates the signal that it receives to every node in the first hidden layer. Each connection is made by an edge, which has a weight, by which the signal is multiplied. The weighted signals received by each hidden node are summed: if the total is larger than some threshold, the hidden node outputs 1, but otherwise 0. Thresholding is realized by a sigmoid function $h(x)$, with the following form:

$$h(x) = \frac{1}{1 + e^{-(x-x_0)}}, \quad (4)$$

where x is the sum of the weighted signals, and x_0 is the threshold. Each node in the first hidden layer then propagates

¹The complete form of the Bessel function is $\Psi_{nm}(r, \varphi) = J_n(r)e^{im\varphi}$, but we only consider the real part in this paper.

its output value to every node in the next layer in the same manner as the nodes in the input layer, and this process is repeated until the output layer is reached.

An ANN can be trained to perform regression or classification. We constructed an ANN to perform regression, in which the output layer consists of one node, as shown in Fig. 8(a). The weighted signals received by this output node are summed and the result is a prediction of the etch bias. We also constructed an ANN for classification, with an output layer consisting of $2N+1$ nodes, each of which corresponds to a certain range of etch bias values, as shown in Fig. 8(b). This ANN maps an input segment to a single output node, which returns 1 while the others return 0; the median of the corresponding range of etch bias values becomes the predicted etch bias of the interesting segment. The range covered by each node has an extent of δ , which is used to determine the convergence criteria in the EPC algorithm in Section II-D.

We evaluated each ANN over all the training segments, and optimized the edge weights and threshold values to minimize the cost, which is the difference between the predicted and the actual etch bias. This minimization is performed by the method of steepest descent, which requires many computations of the derivative of the cost function. However, because the sigmoid function (4) has a simple derivative,

$$\frac{dh(x)}{dx} = h(x)(1-h(x)), \quad (5)$$

the minimization is fast. The resulting ANN is our etch bias model.

The accuracy of prediction is related to the complexity of the ANN. An ANN with insufficient nodes and layers cannot be trained successfully. Conversely, an over-complex ANN may be too specific to the training segments, making it overfitted, and produces poor results for segments with different configurations. We adjust the numbers of nodes and layers through k -fold cross-validation. The set of training segments is randomly divided into k subsets of equal size; $k-1$ of these

Input: A design layout \mathcal{D}_{in}
Output: A litho pattern layout \mathcal{L}_{out}

- L1: $\mathcal{L} \leftarrow \mathcal{D}_{in}$
- L2: **repeat** for max_iterations
- L3: A set of biases $\leftarrow ANN(\text{a set of segments from } \mathcal{L})$
- L4: $\mathcal{D} \leftarrow ETCH(\mathcal{L}, \text{ a set of biases})$
- L5: A set of EPEs $\leftarrow Measure_EPE(\mathcal{D}_{in}, \mathcal{D})$
- L6: **if** $EPE_{max} \leq \epsilon$ **then** Exit loop
- L7: $\mathcal{L} \leftarrow CORRECT(\mathcal{L}, -\alpha \times \text{a set of EPEs})$
- L8: **return** $\mathcal{L}_{out} \leftarrow \mathcal{L}$

Fig. 9. The EPC algorithm.

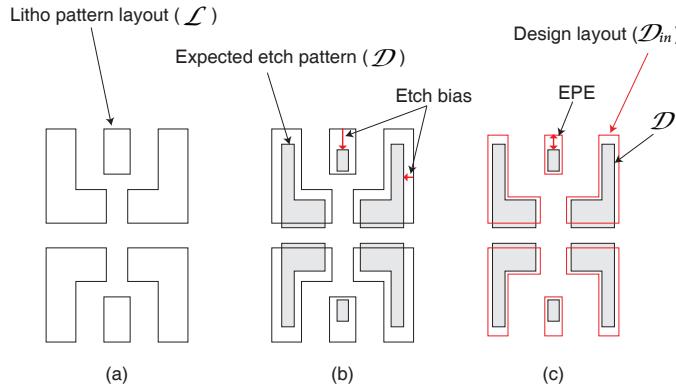


Fig. 10. (a) Current litho pattern layout (\mathcal{L}), (b) expected etch pattern (\mathcal{D}), and (c) EPE measured between the expected etch pattern (\mathcal{D}) and a design layout (\mathcal{D}_{in}).

sets are used to construct an ANN while the remaining set is used to assess the proportion of correctly predicted etch biases. The accuracy is averaged over k iterations, and the whole iterations are repeated for another ANN with different numbers of nodes and layers. This adjustment procedure is embedded into an optimization that determines the numbers of nodes and layers that give the highest accuracy.

Once an ANN has been trained, it is used to predict the etch pattern for a litho pattern layout. A set of parameters is extracted from each segment in the layout, as we described in Section II-B, and submitted to the ANN, which then predicts the etch bias of that segment. The segments are then moved in or out following the bias, which yields a predicted etch pattern.

D. Application of the EPC Algorithm

EPC is to find a litho pattern layout that is expected to produce the desired etch pattern, as shown in Fig. 2. Since no analytical function exist to model etch proximity effect, EPC cannot be performed in analytical fashion, so we have to rely on trial-and-error.

Our EPC algorithm, shown in Fig. 9, receives a design layout \mathcal{D}_{in} , which becomes the initial litho pattern layout \mathcal{L} (L1), and outputs a corrected litho pattern layout \mathcal{L}_{out} that is expected to produce the desired etch pattern. \mathcal{L} is modified iteratively (L2-L7), with the aim of making the expected etch pattern \mathcal{D} as similar as possible to the desired etch pattern \mathcal{D}_{in} .

Each segment of the litho pattern layout \mathcal{L} is parameterized and inputted to the ANN, which returns (L3) a set of expected etch biases for all the segments. Each edge of \mathcal{L} is moved following the expected etch bias, which yields expected etch pattern \mathcal{D} (L4), as shown in Fig. 10(b). The expected etch pattern is then compared with the design layout; the difference between them is modeled by edge placement errors (EPEs), which are signed distances between the edges of \mathcal{D}_{in} and \mathcal{D} , as shown in Fig. 10(c). We measure the EPE at the center of every segment of \mathcal{D}_{in} (L5). A new \mathcal{L} is constructed (L7) by moving each segment of the current \mathcal{L} in the opposite direction to the EPE through a distance of $\alpha \times EPE$, where α is a user-defined coefficient that improves convergence. If the magnitude of the largest EPE is less than a threshold ϵ , then iteration stops; otherwise it continues until a user-defined iteration-count is reached. Note that ϵ needs to be larger than δ , which is the precision of the predicted etch bias obtained from the ANN, as discussed in Section II-C.

III. EXPERIMENTAL ASSESSMENT

ML-EPC was implemented using Proteus [9] for parameter extraction and Python to construct the ANN. We extracted 1200 training segments from both synthetic- and actual-layouts for ANN training; a further 1500 test segments were extracted from the layouts and used to assess accuracy. A segment is represented by 35 parameters: 29 local densities and 6 optical kernel signals, so our ANN has 35 input nodes. We performed 10-fold cross-validation of two alternative ANNs (see Fig. 8(a) and (b)). The ANN for regression had 4 hidden layers, each of 10 hidden nodes, and the ANN for classification had 3 hidden layers, each of 14 hidden nodes. We set $N = 75$ and $\delta = 0.2\text{nm}$ in the ANN for classification, yielding 151 output nodes, to cover etch biases from -15nm to 15nm , with 0.2nm precision.

We implemented a rule-based (RB) and a model-based (MB) EPC. Each rule covers 28 increments in pattern space and 20 increments in pattern width; each of the 560 combinations that result is associated with an etch bias. Two versions of MB-EPC were implemented: MB-2nd was constructed with a version of Eq. (1) that includes all six 2nd-order terms (\mathbf{Vis}^2 and $\mathbf{Vis} \times \mathbf{Den}$ etc.); MB-3rd includes all 3rd-order terms. The size of each kernel and the coefficients were determined through linear regression using Progen [10]. The RB- and MB-EPCs were trained using the same 1200 segments.

A. Accuracy of Etch Bias Model

We compared root mean square (RMS) errors for the 1200 training segments (see white bars in Fig. 11). RB-EPC has very large RMS errors, indicating that a few simple rules cannot represent complicated pattern shapes adequately. With the MB-EPC, the RMS errors decrease as higher-order terms are introduced. Our ANNs for regression and classification respectively produced results with average RMS errors of 2.1nm and 1.6nm , which are placed between MB-2nd (2.3nm) and MB-3rd (1.4nm).

The RMS errors for the 1500 test segments that were not used in model training are also compared (see black bars). RB now produces even larger errors, probably due to its

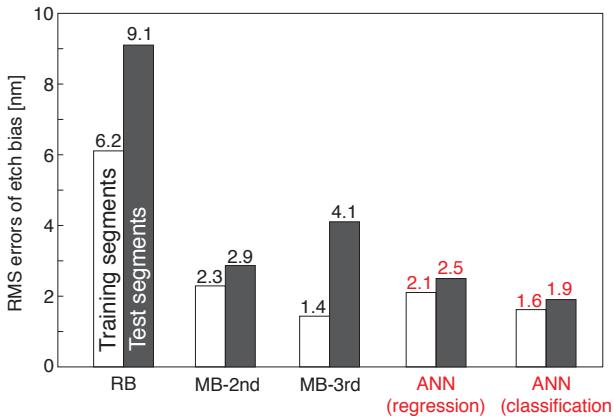


Fig. 11. RMS error of etch bias for training (white) and test (black) patterns.

TABLE I
ERRORS WITH ANNS FOR REGRESSION AND CLASSIFICATION

Segments	RMS errors (nm)			
	Large etch bias (Fig. 11)		Small etch bias	
	Regression	Classification	Regression	Classification
Training	2.1	1.6	0.8	1.1
Test	2.5	1.9	1.2	1.6

poor coverage of complicated patterns. The errors produced by MB-2nd increase only slightly, but the errors from MB-3rd become almost 3 times larger. We attribute this to the inability of empirical kernels to model etching phenomena accurately, which may cause over-fitting when too many terms are involved in model training.

B. Comparison of the Two Types of ANN

In general, using an ANN for classification results in a slightly smaller RMS errors than an ANN for regression, as shown in Fig. 11. However, we observed that the ANN for regression achieves smaller errors than the ANN for classification when the etch bias is small.

We prepared a second set of 1200 training and 1500 test segments with an etch bias ranging from -10nm to 5nm, which is little more than half of its range in the experiments which generated the results shown in Fig. 11. As before, two types of ANN were constructed. This time, the ANN for regression produced smaller errors than classification (see columns 4 and 5 in Table I). This suggests that an ANN for regression may be more suitable for a weak etch process, which produces small etch biases; whereas classification is likely to be more suitable for a strong etch process, which produces large etch biases.

C. Sampling of Training Segments

We prepared a third set of 1200 training segments that are randomly extracted from both the synthetic patterns and actual layout. An ANN for classification was constructed for these segments, and its RMS errors on the training segments was 1.4nm, which is 0.2nm smaller than the results in Fig. 11 because the set of training segments includes a number of

TABLE II
ERRORS WITH DIFFERENT PARAMETER SETS

Parameter sets	# Parameters	RMS errors (nm)	
		Training	Test
Den	49	3.3	3.5
Den+Opt	59	1.5	1.9
Den+Opt (reduced)	35	1.6	1.9

segments with similar surroundings. But the RMS errors for the test segments become 2.5nm, which is 0.6nm larger than the results in Fig. 11, because the randomly sampled segments do not include a range of patterns comprehensively.

D. Changing the Parameter Sets

We tried three different combinations of parameter types with ANNs for classification. Table II shows the RMS errors for the training and test segments classified using the three parameter sets. The significant contribution of optical kernel signal to accuracy is largely independent of that of the local pattern densities. If no optical kernel signals are considered (see the second row), the RMS error is very large for both the training and test segments. Accuracy is much improved when we introduce optical kernel signals (see the third row).

We also tried to reduce the number of parameters by summing pairs of densities in symmetric locations, and using the result as a single parameter; and we only use the 6 most significant optical kernel signals, as we discussed in Section II-B. The number of parameters is in turn reduced by 40% (last row), and the runtime of the ANN is reduced by 49%, with an RMS accuracy of 1.6nm and 1.9nm for the training and test segments respectively.

E. Effect of Cross-validation

To investigate the effect of cross-validation, we constructed two more ANNs for classification. One is a simple ANN, consisting of a single hidden layer with 10 nodes, which produces very high RMS errors for both training (6.7nm) and test segments (7.8nm); this is an understandable consequence of its simplicity. The second ANN is more complicated, with 5 hidden layers, each consisting of 15 nodes. It produces very small RMS errors for training segments (1.1nm) which we would expect with this number of degrees of freedom; but its error with test segments is 2.8nm, which is 0.9nm larger than the errors produced by an ANN constructed using cross-validation. We attribute this decline to over-fitting.

F. Assessment of New EPC

We compared ML-EPC and MB-2nd in terms of the critical dimension (CD); MB-2nd is used for etch bias prediction in L3 of Fig. 9. MB-EPC was applied to a 20nm DRAM gate layer, and the gate CD was measured in the core and peripheral regions. The distribution of CD errors is shown (dashed curve) in Fig. 12, in which the 3-sigma value is 8.2% of the target CD. The same experiment was performed using ML-EPC with the ANN for classification, and the resulting CD error distribution

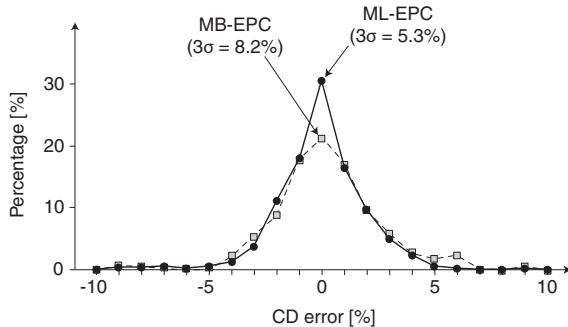


Fig. 12. Histogram of CD errors on a wafer after MB-EPC and ML-EPC.

is shown as a continuous curve, and the 3-sigma value is reduced to 5.3%.

ML-EPC requires about 8 times more runtime than MB-2nd because we extract many local densities around every interesting segment. We expect that runtime can significantly be improved if commercial tools are optimized to measure local densities. Also, our method is currently implemented on top of Proteus [9], using a few user-assessable functions, so is inherently very slow.

IV. CONCLUSIONS

We have presented a new etch bias model based on machine learning. The key components of our model are the methods of preparing training segments and parameterizing them. Training segments are systematically prepared to keep their number small, while ensuring sufficient coverage to handle the shapes of patterns in real layouts. Local density distributions and optical kernel signals have been proposed as parameters, and we reduced their numbers by merging symmetric parameters and selecting the kernels which make the most significant contribution. An artificial neural network (ANN) is used as the etch bias model, with cross-validation to avoid over-fitting. We have applied our method to a 20nm DRAM gate layer. Experiments demonstrated that our method yields predicted etch biases with smaller RMS errors, and gate CDs with smaller OCVs, compared to conventional rule- and model-based EPC.

REFERENCES

- [1] S. Shim and Y. Shin, "Etch proximity correction through machine-learning-driven etch bias model," in *Proc. SPIE Advanced Lithography*, Mar. 2016, pp. 1–10.
- [2] Y. Granik, "Correction for etch proximity: new models and applications," in *Proc. SPIE Advanced Lithography*, Mar. 2001, pp. 98–112.
- [3] Samsung Electronics Corp., OPC Principal Engineer, *personal communication*, May 2016.
- [4] M. Salama and A. Hamouda, "Efficient etch bias compensation techniques for accurate on-wafer patterning," in *Proc. SPIE Advanced Lithography*, Mar. 2015, pp. 1–7.
- [5] J. Park, S. Kim, S. Shim, S. Suh, and H. Oh, "The effective etch process proximity correction methodology for improving on chip CD variation on 20nm node DRAM gate," in *Proc. SPIE Advanced Lithography*, Mar. 2011, pp. 1–10.
- [6] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. Fifth Berkeley Symp. on Mathematical Statistics and Probability*, vol. 1, Dec. 1967, pp. 281–297.

- [7] R. Ng and J. Han, "CLARANS: a method for clustering objects for spatial data mining," *IEEE Trans. on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 1003–1016, Sep. 2002.
- [8] C. Wu, W. Yang, L. Luan, and H. Song, "Photoresist 3D profile related etch process simulation and its application to full chip etch compact modeling," in *Proc. SPIE Advanced Lithography*, Mar. 2015, pp. 1–8.
- [9] Synopsys, "Proteus," Dec. 2013.
- [10] Synopsys, "Progen," Dec. 2013.



Seongbo Shim received B.S. and M.S. degrees in physics from Seoul National University, Seoul, Korea, in 2004 and 2006 respectively, and a Ph.D. from the School of Electrical Engineering, KAIST, Korea, in 2016. From 2006 to 2012, he was with the Semiconductor R&D Center, Samsung Electronics, where he was a Senior Engineer working on computational lithography, OPC, and DFM for advanced technologies. He has authored more than 30 papers on semiconductor manufacturing (lithography and OPC) and DFM. He is the holder of 12 patents on OPC and lithography. His research interests include mask synthesis (OPC) algorithms, DFM, VLSI CAD for the design-manufacturing interface, and design technology co-optimization (DTCO) for emerging technologies.



Youngsoo Shin (M'00–SM'05) received B.S., M.S., and Ph.D. degrees in Electronics Engineering from Seoul National University, Korea. From 2000 to 2001, he was a Research Associate with the University of Tokyo, Japan, and from 2001 to 2004 he was a Research Staff Member with the IBM T. J. Watson Research Center, Yorktown Heights, NY. He joined the Department of Electrical Engineering, KAIST, Korea, in 2004, where he is currently a professor. His current research interests include CAD with emphasis on low-power design and design tools, high-level synthesis, sequential synthesis, and programmable logic. Dr. Shin received the Best Paper Award at ISQED in 2005. He has been a member of the technical program committees and organizing committees of many conferences including DAC, ICCAD, ISLPED, ASP-DAC, ICCD, and VLSI-SoC. He is an Associate Editor of the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems and IEEE Design & Test.