An Efficient Method for Chip-Level Statistical Capacitance Extraction Considering Process Variations with Spatial Correlation

Wangyang Zhang[1], Wenjian Yu[1], Zeyi Wang[1], Zhiping Yu[2], Rong Jiang[3], and Jinjun Xiong[4]
[1]Dept. Computer Science & Technology, Tsinghua University, Beijing 100084, China
[2]Institute of Microelectronics, Tsinghua University, Beijing 100084, China
[3]Cadence Design Systems Inc., San Jose, CA 95131, USA
[4]IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA

Abstract

An efficient method is proposed to consider the process variations with spatial correlation, for chip-level capacitance extraction based on the window technique. In each window, an efficient technique of Hermite polynomial collocation (HPC) is presented to extract the statistical capacitance. The capacitance covariances between windows are then calculated to reflect the spatial correlation. The proposed method is practical for chip-level extraction task, and the experiments on full-path extraction exhibit its high accuracy and efficiency.

1. Introduction

Process variations can be classified into systematic variations and random variations [2]. The systematic variations are often pattern-dependent and can be modeled with some deterministic methods, while the random variations need a stochastic modeling methodology for parasitic extraction. Some characterization methods for pattern dependent variations of parasitic parameters were proposed in [1]. For the random-variation-aware capacitance extraction, a straight-forward approach is the Monte Carlo method, which suffers from huge computing time with thousands of stochastic samplings.

A non-sampling method named FastSies, was proposed in [3] to capture the rough surface effect in capacitance extraction. Based on a 3-D boundary element method (BEM) capacitance solver, a perturbation method was proposed in [4] to generate a quadratic model for the capacitances. A similar work was recently proposed with a spectral stochastic collocation method [5]. Methods in [4, 5] model the variation as the fluctuation of each panel on conductor surface. Although the techniques to reduce the random variables are used, the number of random variables remains very large for problems with short correlation length. Since the computational time is roughly that of running 3-D field solver for k^2 times, where k is the number of random variables, the advantage of methods in [4, 5] over the Monte Carlo method may be limited. Methods in [3-5] are only suitable to model the off-chip rough surface effect, because they only handle surface fluctuation but exclude other variation sources like spacing. Also, these methods can only handle small structures.

In actual chip-level capacitance extraction, only a small window of interconnect structure is simulated with a field solver or table-lookup method [6]. Then, the distributed capacitance model is generated for the following timing analysis or circuit simulation. If the total capacitance of a critical path is needed, the capacitances obtained within the windows related to the critical path need to be summed up in some manner [6]. To consider the process variations in chip-level capacitance extraction, not only the statistical capacitance within the window is needed, but the covariance of capacitances from different windows is required to reflect the spatial correlation of the parameters. Although there are several methods for statistical capacitance extraction, none of them considers the covariance of capacitances for actual chip-level extraction.

In this paper, a simple variation model is adopted to consider the main variation behavior of on-chip interconnects. Based on this model, a technique of Hermite polynomial collocation (HPC) is presented, which generates a second-order stochastic model for capacitance. This method can be implemented on any kind of capacitance solver. To capture the capacitance correlation among the extraction windows, the formula of capacitance covariance is then derived. As an application and extension, the statistical method to calculate the full-path capacitance is proposed. Experiments are carried out with the proposed method and the Monte Carlo method, which shows that the former has more than 100× speedup while preserving high accuracy.

2. Preliminary

2.1 Process Variations with Spatial Correlation

This work is supported by NSFC under Grant 90407004, and in part by the Basic Research Foundation of Tsinghua National Laboratory for Information Science and Technology (TNList).

978-3-9810801-3-1/DATE08 © 2008 EDAA

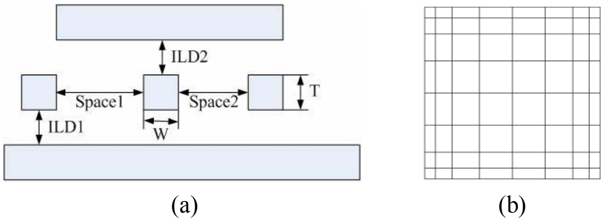


Fig. 1 A practical variation model with (a) multiple variation sources, and (b) a grid setting for variation and spatial correlation.

Multiple variation sources may exist in a practical chip, as shown in Fig. 1(a). Each process variation can be decomposed by systematic and random part. Therefore, for each geometrical parameter F, we have

$$F = F_0 + F_s + F_r \tag{1}$$

where F_0, F_s and F_r denote the nominal value, systematic variation and random variation of F, respectively. In this work, we focus on the effect of random variation. Supposing the systematic variation is estimated with some technique, the parameter F becomes the sum of a nominal value and a zero-mean random variable F_r.

For the random variation, an important feature is spatial correlation [2]. One model for spatial correlation utilizes a set of grid cells superimposed on chip area, as shown in Fig. 1(b). The chip is dissected into grid cells of different sizes, and the random process variation is considered the same within each cell. The way of gridding can be decided with the knowledge of manufacturing process. For example, Fig. 1(b) shows a gridding scheme, where the center grid cells are coarser and less than those at boundary. The random variable F_r may have different variance at different grid cells. And the joint spatial variation of this parameter for all grid cells follows a multivariate Gaussian distribution with a correlation matrix Ω. A method to evaluate the variance and the correlation matrix was proposed in [2].

2.2 Homogenous Chaos Expansion

The homogenous chaos expansion is an efficient way to approximate a stochastic function:

$$C(\xi) = \sum_{j=1}^{\infty} c_j \Psi_j(\xi) \tag{2}$$

where $\xi=(\xi_1, \ldots, \xi_d)$ is a set of Gaussian random variables, and Ψ_j are Hermite polynomials in ascending order. Homogeneous Chaos expansion is guaranteed to converge for any Gaussian random process with finite second-order moments [7]. Moreover, the Askey principle [8] showed that expansion based on Hermite polynomials has the optimal convergence rate for a Gaussian random process.

Approximation using the homogenous chaos expansion needs to truncate the series (2), preserving the first p-order terms. The Hermite polynomials below order 3 are: 1 for order 0, ξ_i (i=1, …, d) for order 1, and ξ_i^2-1 (i=1, …, d), $\xi_i\xi_j$ (i, j=1, …, d, $i≠j$) for order 2.

According to the Galerkin method, the truncation error is minimized when

$$< C(\xi), \Psi_k(\xi) >=< \sum_{j=1}^{M} c_j \Psi_j(\xi), \Psi_k(\xi) >, \; k = 1, \; 2, \; ..., \; \mathrm{M} \tag{3}$$

where the inner product is defined as

$$< X, Y >= E(XY) \;\; , \tag{4}$$

and M is the number of Hermite polynomials Based on the orthogonality of Hermite polynomials, the coefficients can be determined by

$$c_j = \frac{< C(\xi), \Psi_j(\xi) >}{< \Psi_j(\xi), \Psi_j(\xi) >} \qquad . \tag{5}$$

3. Efficient Statistical Capacitance Extraction

Below an efficient algorithm for statistical capacitance extraction is presented.

3.1 Hermite Polynomial Collocation Method

To model capacitance with second-order accuracy, we adopt a Hermite polynomial collocation (HPC) method, which was proved in [5] to have higher accuracy over the Taylor-conversion method in [4]. This method requires no conversion and directly represents desired capacitance C by homogenous chaos expansion, i. e. (2). Eq. (5) is used to calculate the coefficients, where the denominator is a fixed value and the numerator is a d-dimensional integral:

$$< C, \Psi_j >= \int C(\xi) \Psi_j(\xi) f(\xi) d\xi \;\; \cdot \tag{6}$$

Here $f(\xi)$ stands for the probability density function (PDF) of ξ. Numerical integration can be used to calculate (6), resulting in a weighted sum representation:

$$< C, \Psi_j >= \sum_{i=1}^{K} w_i C(\xi^i) \Psi_j(\xi^i) \tag{7}$$

where ξ^i is the ith Hermite-Gaussian integral point.

Once we obtain the $C(\xi^i)$, the statistical capacitance can be calculated with (5) and (7). And $C(\xi^i)$ can be calculated by any existing tool for capacitance extraction without modification, since ξ^i is a known non-statistical quantity. This is a prominent advantage of the HPC method.

The computing time of the collocation method is mainly for doing K times of conventional capacitance extraction, where K is the number of collocation points (i.e. integral points in (7)). Sparse grid quadrature [9] is used as an efficient method to reduce the number of collocation points. One-dimensional sparse grid Θ_1^k equals level k Gaussian quadrature, which uses k+1 points to achieve degree 2k+1 of exactness. The level k sparse grid for d-dimensional quadrature chooses points from set (8):

$$\Theta_d^k = \bigcup_{k+1-d \le |i| \le k} (\Theta_1^{i_1} \times \ldots \times \Theta_1^{i_d}) \qquad , \tag{8}$$

where $|i| = \sum i_j$. And the corresponding weight is:

$$w_{j_{i_1} \ldots j_{i_d}}^{i_1 \ldots i_d} = (-1)^{k-|i|} \binom{d-1}{k-|i|} \prod_m w_{j_{i_m}}^{i_m} \qquad , \tag{9}$$

where $\begin{pmatrix} d-1 \\ k-|i| \end{pmatrix}$ is a combination number and w is the weight for corresponding Gaussian points. The level k sparse grid has degree 2k+1 of exactness [9]. We further prove Theorem 1 as our theoretical basis for choosing the sparse grid level.

Theorem 1 *Sparse grid of at least level k is required for an order k representation.*

Proof. The approximation contains order k polynomials for both C(ξ) and $\Psi_j(\xi)$ for some j, so there exists $C(\xi)\Psi_j(\xi)$ with order 2k, which requires sparse grid of at least level k with degree 2k+1 of exactness.

Therefore, level 2 and level 1 sparse grid are required for quadratic and linear model, respectively. The number of collocation points is about 2d and $2d^2$ for linear and quadratic model. The time cost is about the same as the Taylor-conversion method, while keeping the accuracy of homogenous chaos expansion.

We summarize several accelerating techniques as follows:

1. When d is too small, the number of collocation points for sparse grid may be larger than that of direct tensor product of Gaussian quadrature. For example, if there are only 2 variables, the number is 5 and 14 for level 1 and 2 sparse grid, compared to 4 and 9 for direct tensor product. In this case, the sparse grid will not be used.
2. The set of sparse grid points (8) may contain the same points with different weights. For example, the level 2 sparse grid for 3 variables contain 4 instances of the point (0,0,0). Combining these points by summing the weights reduces 3 times of capacitance extraction.
3. If a 3-D capacitance solver employing an iterative linear equation solver is used, the process can be accelerated by traversing a minimum spanning tree of collocation points according to their spatial distance, and then using the preceding solution as the initial guess for the next capacitance solution [5].

After solving the coefficients of the Hermite polynomial expansion with (5) and (7), each capacitance C_k inside window i can be represented by

$$C_{ki} = \sum_{j=1}^{M} c_{kij}\Psi_j(\xi) \quad , \tag{10}$$

with mean and variance calculated with

$$\begin{cases} E(C_{ki}) = c_{ki1} \\ D(C_{ki}) = \sum_{j=2}^{M} c_{kij}^2 < \Psi_j(\xi), \Psi_j(\xi) > \end{cases} . \tag{11}$$

3.2 Variable Preprocessing

Variables that belong to the same variation source but located in different variation grid cells are correlated, while the collocation method requires a set of uncorrelated variables. Therefore, a preprocessing step needs to be performed to transform the variables.

Theorem 2 *For a set of Gaussian variables ξ with covariance matrix Δn, If Δn=LL^T, then ξ=Lξ*, where ξ* is an independent set of N(0,1^2).*

Proof.
$$\begin{aligned} \mathrm{cov}(L\xi^*) &= E(L\xi^*(L\xi^*)^T) = LE(\xi^*\xi^{*T})L^T \\ &= LL^T = \Delta n = \mathrm{cov}(\xi) \end{aligned} \tag{12}$$

Cholesky factorization can be performed to decompose the covariance matrix, which is always symmetric positive semidefinite in nature.

Therefore, in the preprocessing step, each set of variables belonging to the same variation source is decomposed using Cholesky factorization, yielding lower-triangular matrix L_i. Then, the overall decomposition matrix L can be formed by diagonally aligning each L_i as a block-diagonal matrix.

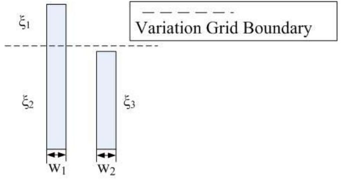


Fig. 2 An example for variable preprocessing.

Fig.2 gives an example for a simple layout. The variation sources are the width of two parallel conductors, and the gridding yields three variables. Suppose w_1, w_2 have standard deviations σ_{w1} and σ_{w2}, ξ_1 and ξ_2 have correlation coefficient ρ. Cholesky decomposition yields:

$$\sigma_{w1}{}^2\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} = LL^T, L = \sigma_{w1}\begin{bmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{bmatrix} \tag{13}$$

and therefore

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} = \begin{bmatrix} \sigma_{w1} & 0 & 0 \\ \sigma_{w1}\rho & \sigma_{w1}\sqrt{1-\rho^2} & 0 \\ 0 & 0 & \sigma_{w2} \end{bmatrix} \begin{bmatrix} \xi_1^* \\ \xi_2^* \\ \xi_3^* \end{bmatrix} \tag{14}$$

Therefore, the algorithm for statistical capacitance extraction within window is as follows:

Algorithm *Intra-Window Capacitance Extraction* ***(W$_i$)***

```
1.  Preprocess variables inside $W_i$
2.  Calculate collocation points $\{p_j\}$
3.      For each $p_j$
4.          Solve desired capacitance in $W_i$ at $p_j$
5.      For each desired capacitance k
6.          Evaluate capacitance $C_{ki}$ and variance $D_{ki}$
```

4. Chip-Level Capacitance Extraction Considering Spatial Correlation

4.1 Window and Grid Partition

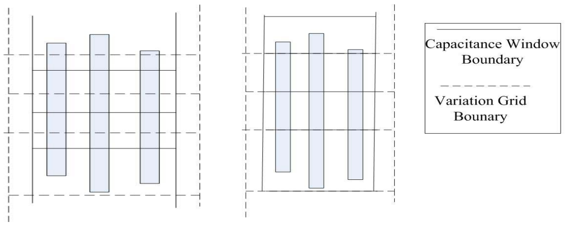
For chip-level extraction, the window technique is necessary to limit the problem size. Only the small structure within a window is simulated with a capacitance solver. Then, the distributed capacitance elements are obtained. For a critical net, the full-path extraction may be needed, which utilizes the capacitances from related windows to assemble the total and coupling capacitances of the whole net. Sophisticated techniques were proposed in [6] for window partition and capacitance assembling.

Since here we focus on the variation-aware capacitance extraction, we only consider a simple window technique. The windows are partitioned to be non-overlapping, and the whole-net capacitance is simply the sum of related window capacitances. This assumption does not prevent our method from utilizing other windowing techniques.

The variation grid defines the variation model. The spatial correlation of same physical variable in different grid cells is usually calculated with the formulae [2]:

$$\rho = \exp(-r^2 / \eta^2) \tag{15}$$

where r is the distance between two spatial positions, and η is called correlation length. Large correlation length indicates strong correlation between grid cells close to each other, and vice versa. Since the variation grid is independent from the extraction window, their spatial relation can be two kinds, shown in Fig. 3(a) and Fig. 3(b). If the windows and grid overlap, the extraction window will involve more variables. Otherwise, there are fewer variables if their boundaries coincide with each other.



(a) (b)

Fig.3 Two kinds of window and grid configuration.

4.2 Inter-Window Covariance of Capacitance

Several methods can be used for the variation-aware capacitance extraction in a window, among which the HPC in Section 3 is an efficient one. However, the existing methods do not consider the relation of intra-window statistical capacitances caused by the spatial correlation of parameters. The covariance of capacitances of two windows is needed. Suppose the intra-window capacitance is represented by (10), the covariance of capacitances can be evaluated according to the linearity of covariance:

$$\mathrm{cov}(C_{ki}, C_{kj}) = \sum_{p=1}^{M_i} \sum_{q=1}^{M_j} c_{kip} c_{kjq} \,\mathrm{cov}(\Psi_p(\xi), \Psi_q(\xi')) \tag{16}$$

where C_{ki} and C_{kj} stand for capacitances of net k in window i and j. They are represented by M_i and M_j functionals, respectively. ξ and ξ' are the variable set in windows i and j, respectively. Now, the problem becomes calculating the covariance between homogenous chaos functionals.

The first step is to determine the correlation between ξ and ξ', which can be achieved by reverting them to the physical parameters before preprocessing:

$$\xi = L_i^{-1}\xi^*, \xi' = L_j^{-1}\xi'^* \tag{17}$$

For any two variables $\xi_a \in \xi$, $\xi'_b \in \xi'$, correlation exists only if they belong to the same variation source. With (17),

$$\begin{aligned}\mathrm{cov}(\xi_a, \xi'_b) &= \mathrm{cov}(\sum_t l_{iat}\xi_t^*, \sum_r l_{jbr}\xi_r'^*) \\ &= \sum_t \sum_r l_{iat} l_{jbr} \,\mathrm{cov}(\xi_t^*, \xi_r'^*)\end{aligned} \tag{18}$$

where l_{iat} and l_{jbr} are elements of L_i^{-1} and L_j^{-1}. $\mathrm{cov}(\xi_t^*, \xi_r'^*)$ can be directly checked out from the correlation matrix.

By definition, there are 4 types of functionals $\Psi(\xi)$ below order 3. The level 0 functional can be neglected since it has covariance 0 with any functional. The remaining 3 types yield 6 different kinds of covariance pairs in (16), which are listed as follows:

$$\mathrm{cov}(\xi_a, \xi'_b) \tag{19}$$

$$\mathrm{cov}(\xi_a{}^2 - 1, \xi_b'^2 - 1) = 2\,\mathrm{cov}(\xi_a, \xi'_b)^2 \tag{20}$$

$$\mathrm{cov}(\xi_a\xi_c, \xi_b'^2 - 1) = 2\,\mathrm{cov}(\xi_a, \xi'_b)\,\mathrm{cov}(\xi_c, \xi'_b) \tag{21}$$

$$\begin{aligned}&\mathrm{cov}(\xi_a\xi_c, \xi'_b\xi'_d) = \\ &\quad \mathrm{cov}(\xi_a, \xi'_b)\,\mathrm{cov}(\xi_c, \xi'_d) + \mathrm{cov}(\xi_a, \xi'_d)\,\mathrm{cov}(\xi_c, \xi'_b)\end{aligned} \tag{22}$$

$$\mathrm{cov}(\xi_a\xi_c, \xi'_b) = 0 \tag{23}$$

$$\mathrm{cov}(\xi_a{}^2 - 1, \xi'_b) = 0 \tag{24}$$

We only prove (22), and others can be derived in a similar manner. The covariance matrix of the multivariate Gaussian distribution (ξ_a, ξ_c, ξ'_b, ξ'_d) has the form

$$\Delta n = \begin{bmatrix} I & P^T \\ P & I \end{bmatrix}, \ \textit{where}\ P = \begin{bmatrix} \mathrm{cov}(\xi_a, \xi'_b) & \mathrm{cov}(\xi_c, \xi'_b) \\ \mathrm{cov}(\xi_a, \xi'_d) & \mathrm{cov}(\xi_c, \xi'_d) \end{bmatrix} \tag{25}$$

Performing Cholesky decomposition yields

$$\Delta n = LL^T,\ L = \begin{bmatrix} I & 0 \\ P & L_x \end{bmatrix},\ L_x = \begin{bmatrix} x_{11} & 0 \\ x_{21} & x_{22} \end{bmatrix}, \textit{and}\ L_x L_x^T = (I - P^2)\cdot \tag{26}$$

Therefore, according to Theorem 2, we have

$$\begin{aligned}&\mathrm{cov}(\xi_a\xi_c, \xi'_b\xi'_d) \\ &= \mathrm{cov}(\xi_a\xi_c, (P_{11}\xi_a + P_{12}\xi_c + x_{11}\xi_b'^*)(P_{21}\xi_a + P_{22}\xi_c + x_{21}\xi_b'^* + x_{22}\xi_d'^*)) \\ &= \mathrm{cov}(\xi_a\xi_c, (P_{11}P_{22} + P_{12}P_{21})\xi_a\xi_c) \\ &= \mathrm{cov}(\xi_a, \xi'_b)\,\mathrm{cov}(\xi_c, \xi'_d) + \mathrm{cov}(\xi_a, \xi'_d)\,\mathrm{cov}(\xi_c, \xi'_b)\end{aligned} \tag{27}$$

Due to the independence of variation source, we have:

Theorem 3 *The covariance has non-zero value only if the pair satisfies one of the following two conditions:*

a) For (19)-(22), when involved variables are from the same variation source;

b) For (22), when (ξ_a, ξ'_b) and (ξ_c, ξ'_d) are of the same variation source respectively, or (ξ_a, ξ'_d) and (ξ_c, ξ'_b) are of the same variation source respectively.

If the number of grid cells overlapped by each window is sufficiently small, only $O(M_i+M_j)$ calculations need to be performed in (16), and the cost is negligible compared to the extraction process. An acceleration technique can be used to reduce the number of windows involved. With (15), we know that the correlation coefficient decays to about 10^{-4} when the distance between two grid cells is 3η. Therefore, discarding the window pairs with distance more than 3η only results in less than 0.1% error.

4.3 Full-Path Capacitance Extraction

A direct application of the inter-window covariance is the full-path capacitance extraction. With our simple assumption of windowing technique, the mean and variance of capacitance of net k is calculated with:

$$E(C_k) = E(\sum_i C_{ki}) = \sum_i E(C_{ki})$$
$$D(C_k) = D(\sum_i C_{ki}) = \sum_i D(C_{ki}) + 2\sum_{i \neq j} \operatorname{cov}(C_{ki}, C_{kj}) \qquad (28)$$

Therefore, substituting the covariances calculated with (16), we obtain the variance of whole-net capacitance.

If the explicit quadratic form or even the PDF of capacitance is needed, the decomposition (12) must be performed on the covariance matrix for every variation source to gain a uniform basis {δ} for all variables:

$$\xi_t = l_t^T \delta\,, \quad \textit{for any } t \qquad (29)$$

If variation source v has p_v variables, the length of δ is Σp_v. The principle factor analysis (PFA) can be used to reduce the length [4]. Eigen-decomposition on the covariance matrix yields:

$$\Delta n = LL^T, \quad L = (\sqrt{\lambda_1} e_1, \ldots, \sqrt{\lambda_n} e_n) \qquad , \qquad (30)$$

where $\{\lambda_i\}$ are eigenvalues in order of descending magnitude, and $\{e_i\}$ are corresponding eigenvectors. PFA truncates L using the first K terms.

The error introduced by PFA may be very small for the case with large correlation length. For example, for a line crossing 20 grid cells, the error of PFA is only 1% while the number of variables reduces to 5, if the correlation length is 10 times of size of grid cell. But if the correlation length is only two times of that, PFA acts merely as a full decomposition, and the low cost Cholesky decomposition becomes preferable.

Subsituting (29) into (10), we get an expression:

$$C = C_0 + a\delta + \delta^T A \delta \qquad . \qquad (31)$$

Since capacitances for all windows are represented using a same basis, the quadratic form of full-path capacitance can be calculated by directly summing up (31) for all windows without the need to compute covariance. Then, with a technique of characteristic function, the PDF of the full-path capacitance can be obtained [4].

Finally, we give the algorithm for full-path extraction.

Algorithm *Full-Path Capacitance Extraction*

```
1.  Partition windows for capacitance extraction
2.  For each window $W_i$ do
3.      Run Intra-Window Extraction($W_i$)
4.  For each critical net $k$ with related window set $W^k$
5.      Utilize the capacitances for windows in $W^k$ to
        calculate the full-path capacitance.
```

5. Numerical Experiments

Several interconnect structures with window partition are tested with the proposed method for the statistical full-path capacitance. A net crossing all windows is considered as a critical path, whose capacitances are calculated. For simplicity, the capacitances from different windows are simply summed up to get the total and coupling capacitances. The Monte Carlo (MC) simulation with 10000 samples is performed for comparison, as that in [4]. The FastCap 2.0 [10] is used to extract the capacitances for each window. When calculating the PDF of capacitance, the PFA is not used. All experiments are carried out on a Sun V880 server with 750 MHz CPU.

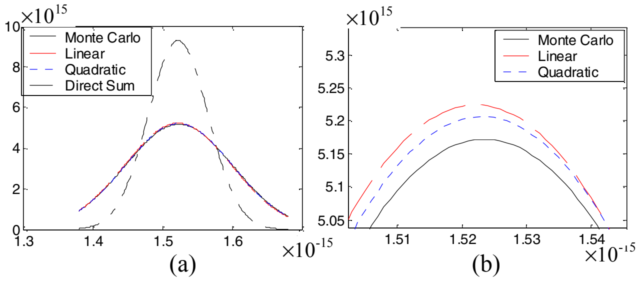
The first case, containing two parallel lines dissected into 10 windows, is used to demonstrate the accuracy and efficiency of our method. Each line is of 40μm length and 1μm width and height, and the spacing of line is 2μm. The variation sources are line height and width. For each window, there are 3 variables, one for height and two for width. The standard deviation of variable is set to 0.2μm. Spatial correlation exists between same kind of variable in different windows, and the correlation length in (15) is set to 8μm. The variation grid is assumed to coincide with the window boundaries, like that in Fig. 3(b). Both linear and quadratic capacitance models are calculated with our method, and their errors on the mean and variance of the capacitances are listed in Table 1. The computing time for each model and the number of collocation points in HPC method for extracting each window are also listed.

Table 1. Efficiency and accuracy of proposed method for case with height and width variations

Model	Points	Time(s)	Total Cap Err.		Coupling Cap Err.	
			Mean	Std	Mean	Std
Linear	7	9.34	-0.10%	-1.00%	-0.06%	-1.31%
Quadratic	25	33.5	-0.03%	-0.66%	0.04%	-0.70%

From Table 1, we can see that both linear and quadratic models achieve high accuracy. The calculated PDF's of total capacitance are shown in Fig. 4(a), with an enlarged peak view shown in Fig. 4(b). The full-path capacitance without considering the spatial correlation is also depicted in Fig. 4(a), with label of "Direct Sum". It is obvious that the capacitance variance is remarkably underestimated if spatial correlation is not considered.

Since the point number in Table 1 equals to the time of invoking FastCap for each window, calculating linear and quadratic models need 70 and 250 runs of FastCap, respectively. In MC simulation, the total time of invoking



(a) (b)

Fig. 4 PDF comparison for the total capacitance.

FastCap is 100000, which is 400 times of that in the proposed method for quadratic model. The actual speed up of our method to MC simulation is 396 (the CPU time of latter is 13293 seconds), which approaches to the ratio of times of invoking FastCap. This is because the computing time for other steps in our method is negligible.

With the same structure, we change the variation grid to get the second test case. Now, the grid cells and windows overlap each other like that in Fig. 3(a). For this case, the number of variables increases to 6 in each window. The computational time and accuracy of both linear and quadratic models are listed in Table 2. The MC simulation time is 14627 seconds. Due to the increase of variable number, the computing time of proposed method rises by 2.0 and 3.8 times for linear and quadratic model, but its speedup to MC is still 762 and 116, respectively.

Table 2. Efficiency and accuracy for case with variation grid cells and windows overlapping

Model	Points	Time(s)	Total Cap Err.		Coupling Cap Err.	
			Mean	Std	Mean	Std
Linear	13	19.2	-0.12%	-0.90%	-0.10%	-1.74%
Quadratic	85	126	0.06%	-0.44%	0.06%	-0.32%

In first two test cases, the linear capacitance model shows enough accuracy, and is 4 times faster than the quadratic model. However, this is not always true. The third test is performed on the same parallel-line case, but with the line spacing reduced to 1μm and chosen as the only variation source. The standard deviation of variable is reduced to 0.1μm accordingly. The computational results are listed in Table 3. For this case, the superiority of quadratic model on accuracy becomes significant. This reflects the non-linear relationship between capacitance and spacing, and the quadratic model is important for densely routed interconnects.

Table 3. Efficiency and accuracy for the case with smaller spacing

Model	Points	Time(s)	Total Cap Err.		Coupling Cap Err.	
			Mean	Std	Mean	Std
Linear	2	2.67	0.04%	-3.45%	0.08%	-3.09%
Quadratic	3	3.97	-0.02%	-0.69%	-0.07%	-0.83%

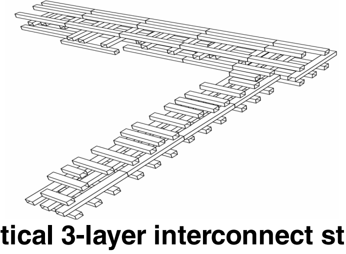


Fig. 5 A practical 3-layer interconnect structure.

Finally, a practical 3-layer structure shown in Fig. 5 is tested. The critical net shifts from the middle layer to the upper layer with an orthogonal turn. This structure includes 8 normal windows with the critical net lying at one layer, and 1 window with the net shifting to the upper layer. The whole size of the structure is about 50μm×50μm. The variation sources are chosen like that shown in Fig. 1(a), and the normal and shift window includes 6 and 10 variables, respectively. The standard deviation is 0.1μm with 8μm correlation length. The computational time and accuracy comparison are listed in Table 4. From it we can see the high accuracy of our method, and its speedup to MC simulation is 718 and 100, for linear model and quadratic model respectively.

Table 4. Efficiency and accuracy for a 3-layer case

Model	Points (normal)	Points (shift)	Time(s)	Speedup to MC	Mean Err.	Std Err.
Linear	13	21	251	718	0.01%	-1.89%
Quadratic	85	221	1803	100	-0.07%	-0.83%

6. Conclusion

A practical framework for chip-level extraction considering spatial correlated variations is proposed. An efficient HPC technique is presented for extract the statistical capacitances within the extraction window. While considering the spatial correlation, the formula for the covariance of capacitances from different windows is derived. Efficient algorithms are also proposed to calculate the statistics of full-path capacitance. Numerical experiments show that our method is of high accuracy and more than 100× faster than the Monte Carlo simulation.

References

[1] X. Qi, A. Gyure, et al, "Measurement and characterization of pattern dependent process variations of interconnect resistance, capacitance and inductance in nanometer technologies," in *Proc. GLS-VLSI,* pp.14-18, 2006.

[2] J. Xiong, V. Zolotov and L. He, "Robust extraction of spatial correlation," *IEEE Trans. CAD*, vol. 26, pp. 619-631, 2007.

[3] Z. Zhu and J. White, "Fastsies: A fast stochastic integral equation solver for modeling the rough surface effect," in *Proc. ICCAD*, pp. 675–682, 2005.

[4] R. Jiang, W. Fu, et al, "Efficient statistical capacitance variability modeling with orthogonal principle factor analysis," in *Proc. ICCAD,* pp. 683-690, 2005.

[5] H. Zhu, X. Zeng, et al, "A sparse grid based spectral stochastic collocation method for variations-aware capacitance extraction of interconnects under nanometer process technology," in *Proc. DATE*, pp.1514-1519, 2007.

[6] W. Shi and F. Yu, "A divide-and-conquer algorithm for 3-D capacitance extraction," *IEEE Trans. CAD*, vol. 23, no. 8, pp. 1157-1163, 2004.

[7] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, New York: Springer-Verlag, 1991.

[8] D. Xiu and G.E. Karniadakis, "The Wiener-Askey polynomial chaos for stochastic differential equations," in *SIAM J. Sci. Comput.*, vol.24, no.2, pp. 619-644, Oct. 2002.

[9] E. Novak and K. Ritter, "Simple cubature formulas with high polynomial exactness," *Constructive Approximation*, vol. 15, no.4, pp. 449-522, Dec.1999.

[10] http://www.rle.mit.edu/cpg/.