# A Fully Automated Method to Create Monte-Carlo MOSFET Model Libraries for Statistical Circuit Simulations

Jing Wang, Henry Trombley*, Josef Watts*, Mai Randall and Richard Wachnik

IBM Semiconductor Research and Development Center (SRDC), Hopewell Junction, NY 12533,
*Essex Junction, VT 05452. E-mail:trombh@us.ibm.com. Tel: 802-769-5786 Fax: 802-769-9659

## ABSTRACT

This paper presents a novel method, Sequential Variation Determination (SVD), to efficiently and automatically determine the required variance for compact model parameters such that the statistical compact model produces the required variances for model output parameters. Unlike previous methods for accomplishing this task, SVD detects inconsistency in the variances and model sensitivities presented. If inconsistency are detected, SVD provides both diagnositic information and makes reasonable adjustment and arrives at a solution.

***Keywords***: statistical compact model, semiconductor variation modeling, variability modeling, interdie variation.

## 1 INTRODUCTION

As the MOSFET feature size continues to shrink, device-to-device variations in a MOSFET have an increasingly important impact on circuit design and manufacturing yield. Therefore, it becomes a critical task for compact modeling engineers to develop accurate, Monte-Carlo (MC) models that capture various device variations to allow statistical circuit simulations. To model the global (or 'die-to-die') variations in a MOSFET, a subset of the parameters for the compact model (e.g., BSIM4 [1] or PSP [2]) are selected to represent the variation. For each of these, a Gaussian distribution needs to be enabled and its variance (or tolerance) adjusted so the output of the MC model matches the upper and lower bounds for a number of critical device metrics. If we manually extract the tolerances of MC model parameters, then a MC simulation (e.g., 500-1000 cases) is needed after changing each model parameter to check whether the match between the MC model and the device variation metrics is satisfactory. This 'manual-extraction' method is very time-consuming because each MC simulation may take 2-5 minutes (on a high performance workstation) and many MC runs are needed. Furthermore, the statistical fluctuations in MC simulations from run to run can compromise the accuracy of the MC model library developed with this approach.

To speed up the MC model extraction process, a *Backward Propagation of Variance (BPV)* method was used in [3-5]. However, with potential inconsistencies in device variation metrics and/or model parameter sensitivities, the BPV approach may not always return a valid solution and in practice, manual optimization is often needed when creating a MC model library using BPV.

In this paper, we introduce a *novel* ***Sequential Variation Determination (SVD)*** technique that extends the BPV method to guarantee a *best* solution of the MC model for a given set of device variation specs. In addition, a diagnosis of the variation spec consistency and model parameter sensitivities is provided. Based on this SVD technique, we developed **a fully automated** tool for creating MC model libraries for advanced CMOS technologies. For a given set of device variation specs (e.g., *Tox*, *Vt*, *Ion*, *Idlin* at various geometries), this tool can automatically set the model parameter tolerances to best match all the variation specs in less than 2 minutes.

## 2 SEQUENTIAL VARIATION DETERMINATION

### 2.1 The Method

If the mathematical problem is well formed, i.e. the specified variances of the output parameters and sensitivities of the selected model parameters are consistent, then the BPV method will quickly give a solution and our method will find the same solution. If the problem is not well formed, e.g. two output parameters are dominated by the same one or two model parameters but have inconsistent specified variances, BPV may produce negative variances for some model parameters. This is not physically reasonable.

To avoid this situation, SVD searches for output parameters that are dominated by a single model parameter and uses that constraint to fixed the corresponding model parameter. The problem is then reformulated using the remaining input and output parameters. Because the set of equations is solved sequentially rather than simultaneously, the solution obtained is approximate. Our implementation provides a method of iterating to improve the first pass solution if necessary. In practice, we find that one pass is adequate in most cases and if not, then a second pass will reach a good solution.

At each step a check for negative variance is conducted. Any negative variances are flagged as potential errors and set to zero. Since we cannot implement negative variance in the model, setting them to zero and reformulating the problem produces a better solution than the mathematically exact solution with negative variances. The user reviews the error messages and can take other corrective actions, adjust unreasonable variance targets, choose a different set



NSTI-Nanotech 2012, www.nsti.org, ISBN 978-1-4665-6275-2 Vol. 2, 2012

of model parameters to vary, or refit the model to adjust unreasonable sensitivities.

## 2.2 Mathematical details

Using the Backward Propagation of Variance technique [3-5], the following equation group is obtained,

$$\left(\frac{\partial M_1}{\partial P_1}\right)^2 \cdot \Delta P_1^2 + \left(\frac{\partial M_1}{\partial P_2}\right)^2 \cdot \Delta P_2^2 + \ldots + \left(\frac{\partial M_1}{\partial P_{N_P}}\right)^2 \cdot \Delta P_{N_P}^2 = (\Delta M_1)^2 \quad (1)$$

$$\ldots$$

$$\left(\frac{\partial M_M}{\partial P_1}\right)^2 \cdot \Delta P_1^2 + \left(\frac{\partial M_{N_M}}{\partial P_2}\right)^2 \cdot \Delta P_2^2 + \ldots + \left(\frac{\partial M_{N_M}}{\partial P_{N_P}}\right)^2 \cdot \Delta P_{N_P}^2 = \left(\Delta M_{N_M}\right)^2$$

Here $\Delta M_i$ ($i$=1,...,$N_M$) is the 3σ variation of each device metric (e.g., *Ion*, *Vtsat*, etc.) and $\Delta P_j$ ($j=1,\ldots,N_P$) is the 3σ variation of each MC model parameter ($N_M \geq N_P$). If we try to solve Eq. (1) using matrix inversion ($N_M = N_P$) or the linear-least-square method ($N_M > N_P$), any inconsistencies in device metric variations ($\Delta M_i$) and/or model sensitivities ($\partial M_i/\partial P_j$) may lead to an invalid solution (e.g., $\Delta P_j^2 < 0$). Therefore, SVD is needed.

For SVD, we reformulate the mathematical problem. In order to simplify the identification of dominating parameters, we scale the equations such that the right hand side (RHS) is a vector of 1's. To enable iteration, if required, we replace the variances of the model parameters with the ratios of the variance to an initial guess of the variance as the vector being solved for. We typically use previous models as a guide to obtain a reasonable initial guess for each parameter. If the initial guesses are poor, iteration using the previous results for the initial guess will quickly converge to a good solution. The solution algorithm proceeds as follows.

1) Construct Matrix ***D*** with each element calculated as $d_{i,j} = (\partial M_i / \partial P_j)^2 \cdot (\Delta P_j^{(0)})^2 / (\Delta M_i)^2$, where $\Delta P_j^{(0)}$ is an initial guess of the model parameter variations. Thus Eq. (1) becomes,

$$\begin{pmatrix} d_{1,1} & d_{1,2} & d_{1,3} & \cdots & d_{1,N_P} \\ d_{2,1} & d_{2,2} & d_{2,3} & \cdots & d_{2,N_P} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ d_{N_M,1} & d_{N_M,2} & d_{N_M,3} & \cdots & d_{N_M,N_P} \end{pmatrix} \bullet \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \vdots \\ \lambda_{N_P} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ \vdots \\ 1 \end{pmatrix} \quad (2)$$

where $\lambda_j = (\Delta P_j / \Delta P_j^{(0)})^2$.

2) For each row in Matrix *D*, search for a dominating element, defined as

$$d_{i,k} > 0.95 \cdot \sum_{j=1}^{NP} d_{i,j} \,. \quad (3)$$

If $d_{m,k}$ is identified as the dominating element of row *m*, $\lambda_k$ is calculated using the following equation,

$$\lambda_k = \left(1 - \sum_{j \neq k} d_{m,j}\right) \Big/ d_{m,k} \quad (4)$$

Then we move all terms related to $\lambda_k$ (i.e., $\boldsymbol{d_{i,k} \cdot \lambda_k}$ , $i=1 \ldots NP$) from the left of Eq. (2) to the right side, and remove row *m* from Matrix ***D***. $P_k$ is saved in List ***LP*** and $M_m$ in List ***LM*** for tracking purpose.

3) With Matrix ***D*** at a reduced size, repeat *Step 2*. Note that the RHS of each row becomes

$$\lambda_i = 1 - d_{i,k} \cdot \lambda_k \quad (5)$$

If at any point, the calculated λ is *negative*, then set it to zero and remove the column associated with it from Matrix ***D***. Then send out a *warning* that a negative variance was generated.

4) If Matrix ***D*** finally reduces to a 1x1 matrix, Eq. (2) is considered solved with *SVD* and the model parameter variations are updated as $\Delta P_j = \mathrm{sqrt}(\lambda_j) \cdot \Delta P_j^{(0)}$. If the size of Matrix ***D*** is still larger than one when no *dominating elements* can be found in any rows, then solve the remaining equation using linear-least-square or matrix inversion. However, given the physics-based nature of the standard FET models (BSIM4/PSP), this situation can be avoided with a carefully selected list of MC model parameters.

5) Test for convergence using:

$$\forall i : 0.95 \leq \lambda_i \leq 1.05 \quad (6)$$

If the calulation has not converged, then use the solution as the new vector of initial guesses and repeat the process.

After the automated process completes, the user reviews the log for warnings. A negative $\lambda_j$ means that the variances of the model parameters set up to that point in the SVD process, caused one or more output parameters to have more variance than specified. Possible reasons include inconsistent specs for output parameter variances, poor choice of model parameters, and excessive sensitivities due to poor fitting of the nominal model.

# 3 ILLUSTRATIVE EXAMPLE

The series of matrices below show two sequential steps of the algorithm for a MOSFET to illustrate how the algorithm proceeds.

| | MCOV | LOV | TOXO | VFBO | UO | RSW1 |
|---|---|---|---|---|---|---|
| Idlin | 0.000021 | 0.000083 | 0.399921 | 0.250455 | 0.339129 | 0.007472 |
| Vtlin | 0 | 0.000002 | 0.156025 | 0.812252 | 0.006202 | 0 |
| Vtsat | 0.005751 | 0.019367 | 0.046584 | 0.36 | 0.003117 | 0.000011 |
| Tinv | 0 | 0 | 1.44 | 0 | 0 | 0 |
| Idsat | 0.004218 | 0.014498 | 0.148032 | 0.149657 | 0.055983 | 0.069135 |
| Cov | 0 | 0.610233 | 0.143791 | 0 | 0 | 0 |

Table 1: Initial **Matrix *D*** for a MOSFET.

The model parameters are all PSP parameters except MCOV which is a parameter in our statistical model representing the correlation between overlap capacitance and effective channel length.

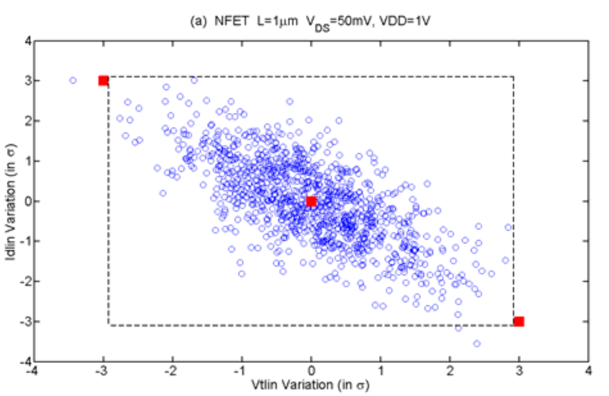
The model parameter TOXO dominates the variance of the output parameter Tinv. The tolerance for TOXO is calculated using (4) and the highlighted row and column are eliminated. The RHS is recalculated using (5). Physically this represents reducing the target variance for each output parameter by the amount of variance accounted for by the just determined Tox tolerance.

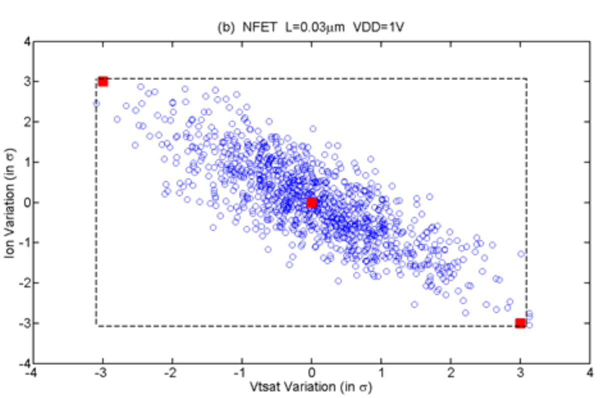
| | MCOV | LOV | VFBO | UO | RSW1 |
|---|---|---|---|---|---|
| Idlin | 0.000021 | 0.000083 | 0.250455 | 0.339129 | 0.007472 |
| Vtlin | 0 | 0.000002 | 0.812252 | 0.006202 | 0 |
| Vtsat | 0.005751 | 0.019367 | 0.36 | 0.003117 | 0.000011 |
| Idsat | 0.004218 | 0.014498 | 0.149657 | 0.055983 | 0.069135 |
| Cov | 0 | 0.610233 | 0 | 0 | 0 |

Table 2: **Matrix *D*** after the first step.

After eliminating TOXO, the model parameter LOV dominates the output parameter Cov. The tolerance for LOV is now calculated and the matrix is reduced again. The process proceeds as described above until all $\lambda$'s have been determined.

## 4 RESULTS



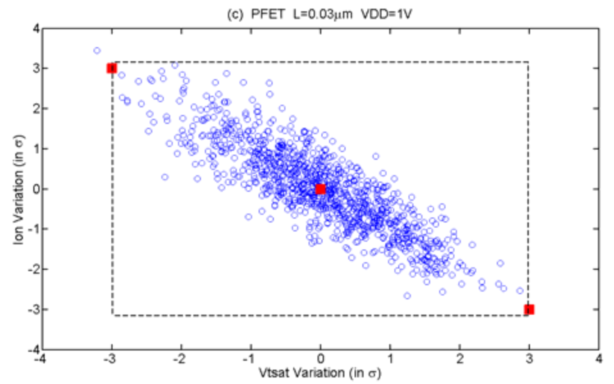




**Fig. 1** Comparison of the MC model results (open circles, 1000 cases) vs. the device specs (closed squares) at the nominal, lower-bound, and upper-bound, respectively. All the numbers are normalized by the σ values defined in the device variation specs. The dashed rectangle in each plot mark the +/-3σ variation of the simulation results.

Fig. 1 shows the simulated results of a MC model library created by our method vs. the device variation specs that were extracted from the hardware data of our 28nm low-power CMOS technology [6]. Our results show that *without any manual adjustments*, this automatically generated MC model library accurately matches all the device variation specs for both nFET and pFET at different geometries.

## 5 CONCLUSION

The sequential variation determination method quickly and automatically solves for the model paramter tolerances required to for a Monte Carlo model to match the specified tolerances on a set of measured device characteristics. The solution by construction has no negative variances and matches all of the device parameter tolerances. If the exact mathematical solution includes negative variance, then the algorithm identifies them to the users so corrections can be made if required.

## REFERENCES

[1] BSIM4 User Manual (URL: www-device.eecs.berkeley.edu/bsim3/bsim4_get.html).
[2] PSP103 User Manual (URL: pspmodel.asu.edu/psp_documentation.htm).
[3] C. C. McAndrew, et al., *BCTM*, p.28, 1997.
[4] C. C. McAndrew, *Proc. IEEE Int. Symp. Quality Electron. Design*, p. 357, 2003.
[5] X. Li, et al., *IEEE T-CAD*, v. 29, n. 4, p. 599, 2010.
[6] F. Arnaud, et al., *IEDM*, Sec. 28-2, p. 651, 2009.