

Application Specific Worst Case Corners using Response Surfaces and Statistical Models

Manidip Sengupta, Sharad Saxena, Lidia Daldoss, Glen Kramer, Sean Minehane, Jianjun Cheng
PDF Solutions, 101 W. Renner Road Suite 325, Richardson, TX 75082

E-mail: {sengupta, saxena, lidiad, gmkrmer, minehane, jianjun}@pdf.com

Abstract

Integrated circuits have to be robust to manufacturing variations. This paper presents a new statistical methodology to determine the worst-case corners for a set of circuit performances. Our methodology first estimates response surfaces for circuit performances as quadratic functions of the process parameters with known statistical distributions. These response surface models are then used to extract the worst-case corners in the process parameter space as the points where the circuit performances are at their min/max values corresponding to a specified statistical level. Corners in the process parameter space close to each other are clustered to reduce their number, which reduces the number of simulations required for design verification. We introduce the novel concept of relaxation coefficient to ensure that the corners capture the min/max values of all the circuit performances at the desired statistical level. The corners are realistic since they track the multivariate statistical distribution of the process parameters. Expected worst-case circuit performances can thus be extracted with a small number of simulations suitable for subsequent design verifications. The methodology is demonstrated with examples showing extraction of corners from digital standard cells and also the corners for analog/RF blocks found in typical communication ICs.

1. Introduction

Random fluctuations of the manufacturing process parameters due to the non-uniformity of the processing steps and environmental operating conditions cause circuit performances to spread around their nominal values. Integrated circuits need to be robust with respect to such statistical variations to avoid parametric yield loss.

Traditional design methodology uses worst-case models in circuit simulations to verify whether circuit performances values are acceptable at all the corners. This methodology is typically used in the area of digital cell design. However, an equivalent methodology does not exist for analog design. Moreover, these methods ignore the correlation among the process parameters and result in a pessimistic design. Statistical device models

alleviate the problem by expressing device performances as functions of a set of statistically independent process parameters, calibrated to reflect the process variations. This allows the distribution of performances to be estimated using Monte-Carlo simulations. However, Monte-Carlo simulations are computationally extremely expensive, especially for large circuits. One method of reducing the computational requirements of Monte-Carlo simulations is to perform design verification at a set of corners, which are expected to represent the conditions that result in worst-case performances.

Many techniques for statistical modeling and simulation have been proposed in the past to extract worst-case corners [1]-[10]. Kang et al. [2] formulated worst-case extraction as a nonlinear programming problem in which the objective is to maximize the joint probability density of the process parameters. In [10] the worst-case vector of the process parameters is chosen such that it causes the worst performances and has the smallest probabilistic distance from the nominal process parameter vector. These approaches do not exploit circuit-level correlations in the process space, and further aggravate the pessimism of traditional worst-case simulations. This issue is addressed in [1], [6] and [8] where circuit primitives are grouped according to their correlation by different techniques. Guardiani et al. [1], [6] proposed to use performance correlation measured from extensive MC simulations of cell library timing performance to identify clusters of percentile points corresponding to a predefined probability value. Maximum likelihood estimation is applied in the process parameter space in order to identify a unique SPICE model for the representative point in every cluster. However, this methodology assumes normal distribution for every circuit performance, and cannot identify the corners for performances with a non-Gaussian distribution.

In this paper we discuss a novel approach to extract the worst-case corner models to capture min/max values for the performances of a selected set of circuits, chosen to reflect the application/product that the device models are being developed for. The distinguishing features of the methodology proposed in this paper are:

- (1) A new algorithm for estimating the values of device model parameters (called process parameters) that result in the desired performance values, and
- (2) Introduction of the concept of a relaxation coefficient to ensure better accuracy in capturing min/max values of all the circuit performances in a multidimensional process parameter space.

Section 2 describes the proposed methodology and its mathematical foundation. In Section 3, we illustrate the results from some of the Figure of Merit circuits where the methodology has been successfully applied. Finally, we conclude the paper with a few comments on this work.

2. Derivation of the Corners

The key steps of our methodology are a) estimating second order (quadratic) Response Surface Methodology Models (RSM) of the circuit performances in terms of the process parameters; b) using the RSM models to estimate the process parameter values that will result in the performance values at the desired statistical level, and c) clustering the parameters to minimize the number of corners, while ensuring that they achieve the desired performance tolerances. The subsections below describe these steps.

2.1. Response Surface Methodology Models

Response surface for a circuit performance is a polynomial approximation of the circuit performance over the range of process parameters of interest. It provides a fast surrogate for the time-consuming circuit simulation over this range. Typically, quadratic RSM models are sufficient to capture the changes in the performances of most digital standard cells and analog circuit blocks over the typical ranges of process parameters. Such models can be written in the form:

$$y_k = a_k + b_k'x + x'B_kx, \quad k = 1, 2, \dots, m$$

for m performances, where for the k -th performance, a_k is a constant term, b_k and B_k are matrices of appropriate dimensions, x the vector of n independent and identically distributed process parameters with a standard normal joint probability density function given by $N(0, I)$. In the following subsections, we will develop a methodology of finding a small set of points in the process parameter (x) space to represent the minimum and maximum values of all the m circuit performances.

2.2. Performances at a particular tolerance level

In this section we find the minimum and maximum values of the performance y_k at the user specified tolerance level, defined as the probability of getting a performance value larger than the maximum value covered by the corners and less than the minimum value covered by the corners. A common method of specifying the tolerance is in terms of the number of equivalent standard deviations (σ) of a standard normal distribution. Let this level be z . From the properties of multidimensional normal density function, the joint probability density function of the process parameters will yield a standard normal distribution when integrated along any straight line in this space. Assuming $y(x)$ is monotonic in this range, the minimum and the maximum values of the performance will be reflected in the process parameter space such that the settings of the process parameters that give the desired tolerance of $z\sigma$ will lie on the hyper sphere with a radius of z in the process parameter space.

Therefore, we search for the minimum and the maximum values of $f(x)$ subject to the constraints that the solution will a vector in the Euclidean process parameter space with a length of z . Mathematically, the problem of finding minimum can be formulated as

$$\min\{y_k = a_k + b_k'x + x'B_kx\}, \quad s.t. \|x\| = z$$

Using the principles of Lagrange multipliers to solve the above problem, characteristic equation for the k -th

$$\phi_k(x) = y_k(x) - \lambda g(x), \quad \text{where}$$

$$y_k(x) = a_k + b_k'x + x'B_kx \quad \text{and} \quad g(x) = x'x - z^2$$

performance y_k can be written as

Differentiating the above equation with respect to x and λ , and setting the derivatives to 0, we get the following

$$\frac{\partial \phi_k(x)}{\partial x} = b_k + 2B_kx - 2\lambda x = 0$$

$$\frac{\partial \phi_k(x)}{\partial \lambda} = x'x - z^2 = 0$$

set of equations:

$$2[\lambda I - B_k]x = b_k \Rightarrow x = \frac{1}{2}[\lambda I - B_k]^{-1}b_k$$

From the first set of n equations,

$$x'x = \frac{1}{4}[[\lambda I - B_k]^{-1}b_k][[\lambda I - B_k]^{-1}b_k] = z^2$$

Putting the above value of x in the $(n+1)$ -th equation,

Solving the resulting set of equations is equivalent to solving for λ in

$$\|(\lambda I - B_k)^{-1} b_k\|^2 = 4z^2$$

and substituting the value of λ in the expression for x . For a linear performance model, $B_k = 0$, and the effect of λ is to scale the gradient vector b_k such that x has the length of z , the desired σ level. However, for a general quadratic model, the above equation does not have a closed form solution for λ . To solve for λ , we construct a function given as

$$\varphi_k(\lambda) = \|(\lambda I - B_k)^{-1} b_k\|^2 - 4z^2$$

and search for its roots. This reduces the optimization problem in an n -dimensional space to a simple root finding problem in one dimension. Figure 1 demonstrates the shape of the above function for different circuit performances.

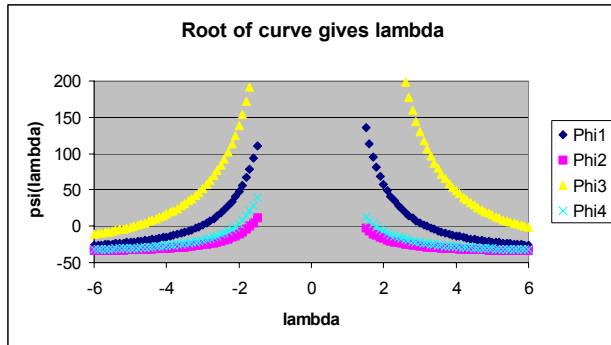


Figure 1. Finding the roots of the curve as λ_1 and λ_2

A line search algorithm over a bracketed pair of values gives the values of λ for individual circuit performances. The effect of B_k can be thought to be disturbing the symmetry of λ_1 and λ_2 , and in general, the two roots of the equation are different in magnitude. In the implementation for the above, we search for the roots of the equation

$$\left\| \left[\lambda I - \sum_k \frac{g_k}{s_k} B_k \right]^{-1} \sum_k \frac{g_k}{s_k} b_k \right\|^2 - 4z^2 = 0$$

We set $g_k = s_k = 1$ to solve for the problem above. The use of the constants g_k and s_k in the implementation will be discussed in the next subsection. The equation is solved iteratively using a standard method of bisection [12] over

two ranges of λ , to find the positive and the negative roots. Once the roots are found, the corresponding x points for the minimum and maximum values of the

$$x = -\frac{1}{2} \left[\lambda I - \sum_{k \in C} \frac{g_k}{s_k} B_k \right]^{-1} \sum_{k \in C} \frac{g_k}{s_k} b_k$$

performance are computed as

These (x) vectors give the minimum and maximum values for the k -th performance such that the tip x lies on the hyper-sphere of radius z in the process parameter space. Note that while using the above implementation to solve the optimization problem, only one term is used in the summation of the scaled matrices in the above equations.

2.3. Clustering the Corners

In the above procedure, we create $2m$ corners for the m circuit performances of interest. Some of these corners are typically close enough in the process parameter space to predict the min/max values of the particular performances they are intended for. The objective is to combine all such corners into groups or clusters and to find a representative point for the group. These representative points are scaled next with the reflection coefficient (α) described in the next subsection.

The clustering methodology employed here is similar to the standard compact clustering technique [13], the difference being in the metric used to define the distance between two clusters of points in the n -dimensional process space. The distance between the points P_i and P_j belonging to two different clusters is defined as

$$\max_{i=1,2,\dots,n_i} \left\{ \max_{j=1,2,\dots,n_j} \{d(i,j)\} \right\}, \quad \text{where}$$

$$d(i,j) = \lim_{r \rightarrow \infty} \left[\sum_{k=1}^{n_{perf}} \left((y_k(x_i) - y_k(x_j)) / (y_{k,max} - y_{k,min}) \right)^r \right]^{1/r}$$

where n_i is the number of points in one cluster and n_j is the number of points in the other one. The performance specific scaling by $(y_{k,max} - y_{k,min})$ is done to normalize the performances of different units, whose numerical values might differ by orders of magnitude. With the above definition, the clustering algorithm can be summarized as:

1. Build $2m$ clusters for the m performances (min/max)
2. While at least 2 clusters are there,
 - (a) Evaluate the scaled distance between each pair of clusters
 - (b) Combine the nearest pair of clusters into a single one

- (c) Evaluate the α levels (see below) for each of the resulting set of clusters.

For each set of clusters, we will have a value of α (defined and explained in the next subsection). This parameter is also used as the stopping criterion for the clustering process. The representative point for each cluster ($\mathbf{x}_{\text{cluster}}$) is determined next by solving the optimization problem

$$\min_x \sum_{k \in C} \frac{g_k}{s_k} [y_k(x) - y_{k,\text{stat}}] \quad \text{subject to} \quad x'x = z^2$$

where $s_k = y_{k,\text{max}} - y_{k,\text{min}}$ is the scaling coefficient for each performance, and the coefficient g_k is defined as 1 if $y_{k,\text{stat}}$ is y_{min} , -1 otherwise. This problem is also solved as a one-dimensional optimization problem using the principles of Lagrange multipliers, as before. The coefficients g_k and s_k can now be used in the same implementation described in the last subsection. For the two roots, we evaluate the objective functions, and the solution (\mathbf{x}) corresponding to the lower value gives us $\mathbf{x}_{\text{cluster}}$.

2.4. Relaxation Coefficient

In the last section, we determined the representative point for each cluster, lying on the hyper sphere of radius z in the process parameter space. In this section, we magnify the vector $\mathbf{x}_{\text{cluster}}$ by a factor α , and term it the relaxation coefficient. Rationale for using such a magnifier is as follows:

Let the cluster under consideration contains a point \mathbf{x}_k that corresponds to the minimum value of the k -th performance. From the optimization procedure in the last section, this is a unique point on the hyper sphere of radius z . Thus, if we represent \mathbf{x}_k by any other point $\mathbf{x}_{\text{cluster}}$ that lies on the same hyper-sphere, the value of y_k will increase. Mathematically,

$$y_k(\mathbf{x}_{k,\text{cluster}}) \geq y_k(\mathbf{x}_k)$$

Consequently, our earlier estimate of the minimum value of y_k may not be reflected by $\mathbf{x}_{\text{cluster}}$. This might lead to a lower range of y_k , during subsequent simulations with the corners. To remedy this effect, we magnify the vector $\mathbf{x}_{\text{cluster}}$ by a cluster magnification factor α such that

$$\mathbf{x}_{k,\text{asc}} = \alpha_k \mathbf{x}_{k,\text{cluster}}, \text{ s.t. } y_{k,\text{asc}} \leq y_{k,\text{min}}$$

Now, as we reduce the number of clusters, the $\mathbf{x}_{\text{cluster}}$ points have to account for the minimum and the maximum of more and more number of performances. As a result, the value of $\alpha = \alpha_{p,\text{max}}$ ($p=1,2,\dots,n_{\text{cluster}}$) will increase. α is the maximum factor by which we magnify

the $\mathbf{x}_{\text{cluster}}$ points to generate \mathbf{x}_{asc} , the application specific corners. As α increases, we slowly deviate from our original constraint of the $z\sigma$ level to define the min/max values of the performances. This factor is thus used as a termination criterion for the clustering process.

The relaxation coefficient makes the approach to building the worst-case corners slightly conservative. For most practical applications, the numerical value of this factor is less than 1.05, which makes the change of coverage of the circuit performance statistically insignificant, especially if the quadratic models used for the performances do not have comparable accuracy. However, it still makes a good indicator to follow, and a sudden jump in its value during the clustering process is used as a guide to terminate the clustering process and to use the current set of clusters as the starting point to define the application specific corners.

There is a trade-off to be considered here. As we reduce the number of corner points for the application under consideration, we change the level of z for which is expected to be the same as the specified tolerance levels of the performances. For example, if we decide on an α level of 1.03 (3% magnification) while determining the 3σ worst-case corners for all the performances, all the tolerances at the 3σ level will be satisfied, but at least a few of the corresponding corner points in the process parameter space will have a deviation of $3 \times 1.03 = 3.09$ from the center of the standard normal distribution. This might result in 3.09σ level worst-case for a few of the circuit performances.

To find the relaxation coefficient (α), we start from $\alpha=1$, and increase it in small steps until $y_k(\alpha \mathbf{x}_{\text{cluster}}) > y_{k,\text{max}}$ (for maximum y_k) or $y_k(\alpha \mathbf{x}_{\text{cluster}}) < y_{k,\text{min}}$ (for minimum y_k). Once the bracket is found, a line optimization is done to determine α_k . This is repeated for all the clusters and the maximum is chosen as α . When there is only a single point in the cluster, $\alpha=1$ and we do not need this step to find the relaxation coefficient.

3. Examples

This section presents examples illustrating the efficacy of our approach in finding realistic worst-case corners. We illustrate the methodology by deriving corners based on standard cells used in digital design and also from analog/RF building blocks of typical communication IC's.

The first example is for standard cells for a 0.13 μm CMOS technology. The experiment performed can be summarized in the following steps:

1. Develop statistical SPICE models to represent the manufacturing variation in device models
2. Select a set of standard cells representative of the applications the process is intended for, and establish their performances. Delays of the standard cells are used for this experiment.
3. Estimate a quadratic RSM for the performances by performing a Resolution V Central Composite Design of Experiment (Res V CCD DOE).
4. Apply the proposed methodology to the performance models (delays of the standard cells) to create the application specific corners.
5. Test if the corners produced encapsulate the drain current distribution of selected devices.

Figure 2 shows the trade off in selection of the number of clusters. We have 10 circuit performances, corresponding to 20 corners. In the clustering process, as we drop to below 10 clusters, we see a significant increase in the α level. Thus, 10 clusters were used to define the corners in this experiment.

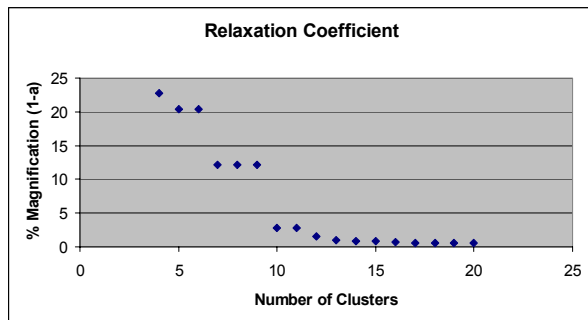


Figure 2. Use of Relaxation Coefficient to determine the number of worst-case corners

Figure 3 demonstrates the efficacy our methodology by showing the location of the 3σ realistic worst-case corners in the space of typical device parameters of interest scatter plots for the N and the P-type device drive currents for a narrow width short channel device, with the σ level for the corners set at 3. Approximately 99.87% of the distribution in the drive currents have been captured by the corners. The “+” signs represent the 4 combinations of the typical “slow” and the “fast” SPICE models. The plots compare the corners with a data set of 3310 observations and Monte Carlo simulation results of a 200 run simulation.

As mentioned in Section I, the slow/fast corners (their 4 combinations) cover a much larger area in the plot. This is also a demonstration of the pessimism the slow/fast device corners in figures 3 and 4.

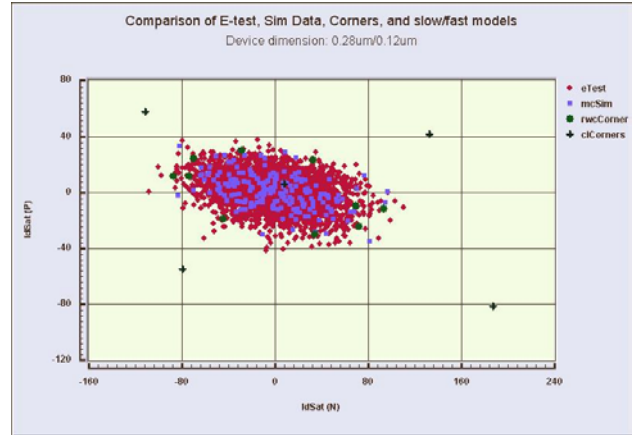


Figure 3. Comparison with the drive current distribution at 3σ level

The methodology is used next to define the corners at the 4.5σ level. In Figure 4, these corners can be seen to define a wider range for the drive current. In both these situations, the Euclidean norms of all the worst-case corner vectors were found to be slightly greater than 3.0 and 4.5 respectively. The differences were less than 3%, which were due to the magnification by the cluster α 's.

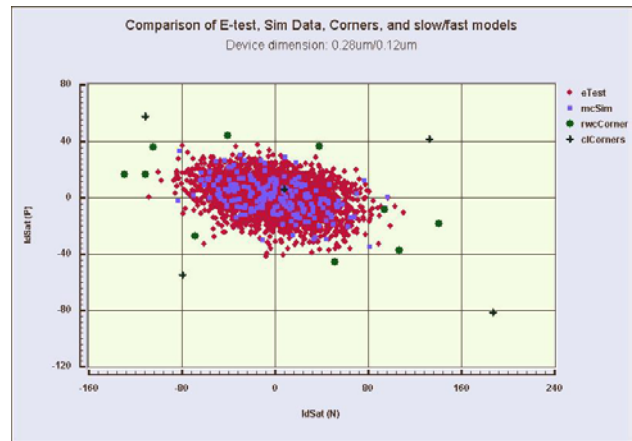


Figure 4. Comparison with the drive current distribution at σ level of 4.5

The next example shows the effectiveness of this technique to find worst-case corners for typical analog/RF blocks found in communication ICs: namely, the noise figure of a low noise amplifier (LNA) and the third harmonic distortion (IIP3) measure of a mixer. Statistical SPICE models for the BJTs used in this design required 11 factors. Six performances are used in the experiment, and after deriving the min/max of each, 5 corners are deemed sufficient to capture all the corners at the 3σ level.

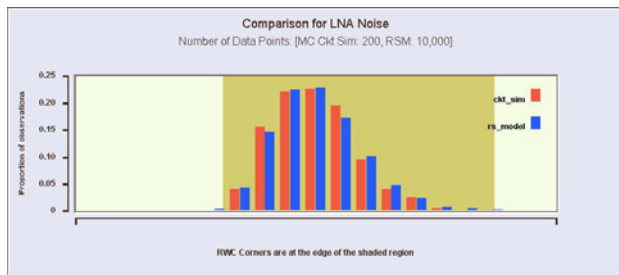


Figure 5. Worst-Case Corners for the Noise Figure of a Low Noise Amplifier (LNA)

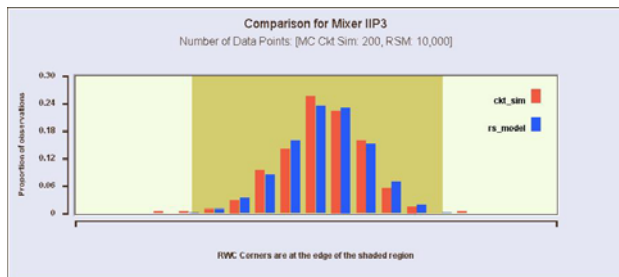


Figure 6. Worst-Case Corners for IIP3 of a Mixer Circuit

Shaded portions in Figures 5 and 6 show how the corners determine the end points of the performance density function. The two histograms compare the performance density functions off Monte Carlo circuit simulation (N=200) results and those off the RSM's (N=10,000). The quadratic performance models are seen to capture the non-Gaussian nature of the performance density function, and the end points are still captured accurately by the corners derived using the proposed methodology.

4. Conclusions

This paper presents a new methodology to generate a minimal set of points in the process parameters space to represent the minimum and maximum values of selected circuit performances in an efficient manner. Quadratic response surface models used for the performances are found to be adequate for modeling variation of digital standard cell performance and key performances of analog/RF components of communication ICs. A new methodology for worst-case corner extraction and corner process parameter clustering and relaxation of the representative point ensures the extraction of accurate corners to capture minimum and maximum performance values in the multidimensional process parameter space.

The simulation time required for statistical analyses is reduced by orders of magnitude with the proposed methodology, from hours or days for the large circuits to a few minutes. Some algorithms used in the implementation could be improved, e.g., use of the

method of bisection [12] to find the roots of an expression for the Lagrange multiplier λ , and the method of LU decomposition [12] for the matrix inversions. However, in the current implementation, the entire process of finding the worst case corners takes a few seconds on Intel x586 family of CPUs using Windows/Linux Operating Systems for both the above examples, with most of the time spent in the clustering process. In summary, application of the proposed algorithm to various circuits like RFIC blocks and standard cell libraries confirm the effectiveness of our approach to find the worst-case corners for a number of different circuit types, selected as a set of representative or Figure of Merit circuits for a specific application.

References

- [1] A. Nardi, A. Neviani, E. Zanoni, M. Quarantelli and C. Guardiani, "Impact of Unrealistic Worst Case Modeling on the Performance of VLSI Circuits in Deep Sub-Micron CMOS Technology", IEEE Transaction on Semiconductor Manufacturing, Vol. 12, N.4, November 1999.
- [2] A.Dharchoudhury and S.M. Kang, "Worst-case analysis and optimization of VLSI circuit performances", IEEE Trans. On Computer-Aided Design, vol. 14, pp.481-492, Apr. 1995.
- [3] M. Bolt, M.Rocchi, J.Engel, "Realistic statistical worst-case simulation of VLSI circuits", IEEE Trans.Semiconduct. Manufact, vol.4, pp.193-198, Aug. 1991.
- [4] K.Singhai and V. Visvanathan, "Statistical device models from worst case files and electrical test data", IEEE Trans. On Semiconduct. Manufact, vol. 12, n.4, pp. 470-484, Nov. 1999.
- [5] S.R. Nassif, A.J. Strojwas and S.W. Director, "A methodology for worst-case analysis of integrated circuits", IEEE Trans. Computer-Aided Design, vol. CAD-5, pp.104-113, Jan. 1986.
- [6] A. Dal Fabbro, B.Franzini, C. Guardiani, "An assigned probability technique to derive realistic worst case timing models of digital standard cells", in Proc. IEEE Design Automation Conf., June 1995, pp.702-706.
- [7] A.N. Lokanathan, J.B. Brockman, "Efficient worst case analysis of integrated circuits", Proc. IEEE Custom Integrated Circuits Conf., May 1995, pp.237-240
- [8] J.C. Chen et al., "Realistic worst-case Spice file extraction using BSIM3, Proc. IEEE Custom Integrated Circuits Conf., May 1995, pp.375-378.
- [9] P. Tuohy, A. Gribben, A.J. Walton, J.M. Robertson, "Realistic worst-case parameters for circuit simulation", IEE Proc., vol. 134, no. 5, pp. 137-140, Oct. 1987
- [10] G.E. Muller-L., "Limit-parameters: the general solution of the worst-case problem for the linearized case", Proc. IEEE Int. Symp. Circuits and Systems, 1990, pp.2256-2259.
- [11] G.E.P. Box, N.R. Draper, "Empirical Model-Building and Response Surfaces, New York, Wiley, 1987.
- [12] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, "Numerical Recipes in C", Cambridge University Press, 1999.
- [13] W.N. Venebles and B.D. Ripley, "Modern Applied Statistics with S", Springer-Verlog, New York, 2002.