

Efficient Design-Specific Worst-Case Corner Extraction for Integrated Circuits

Hong Zhang
Mentor Graphics Corporation
1001 Ridder Park Drive
San Jose, CA 95131
hongzhang@umail.ucsb.edu

Tsung-Hao Chen and Ming-Yuan Ting
Mentor Graphics Corporation
1001 Ridder Park Drive
San Jose, CA 95131
{howard_chen, ming_ting}@mentor.com

Xin Li
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
xinli@ece.cmu.edu

ABSTRACT

While statistical analysis has been considered as an important tool for nanoscale integrated circuit design, many IC designers would like to know the design-specific worst-case corners for circuit debugging and failure diagnosis. In this paper, we propose a novel algorithm to efficiently extract the worst-case corners for nanoscale ICs. Our proposed approach mathematically formulates a quadratically constrained quadratic programming (QCQP) problem for corner extraction. Next, it applies the Lagrange duality theory to convert the non-convex QCQP problem to a convex semi-definite programming (SDP) problem that is easier to solve. Our circuit example designed in a commercial CMOS process demonstrates that the proposed SDP formulation can find the worst-case corners both efficiently and robustly, while the traditional QCQP fails to achieve global convergence.

Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design Aids – Verification

General Terms

Algorithms

Keywords

Process Variation, Worst-Case Corner, Integrated Circuit

1. INTRODUCTION

As IC technologies scale to 45nm and below, integrated circuits are significantly impacted by large-scale process variations, including both inter-die and intra-die variations [1]. Inter-die variations model the common/average variations across the die, while intra-die variations model the individual, but spatially correlated, local variations (e.g., random device mismatches) within the same die. To achieve high parametric yield and, hence, low manufacturing cost, it is of great importance to statistically model and analyze inter-die/intra-die variations for integrated circuits.

To address this need, extensive researches have been performed during the past two decades. As one of the most important techniques, response surface modeling (RSM) was widely applied to statistical circuit analysis and optimization [2]-[11]. Recently, it has been demonstrated that linear RSM is not sufficiently accurate to capture the large-scale variations that are observed for nanoscale technologies; instead, quadratic or even

strongly nonlinear RSM is required to improve modeling accuracy [2]-[6]. Once RSM is created, it can be used to efficiently predict performance distribution and/or parametric yield [7]-[9].

While yield prediction is a necessary step for design sign-off, calculating the yield value only does not meet the needs of circuit designers. If a circuit fails the yield specification, it is important for the CAD tool to provide additional information that helps circuit designers to improve yield. Towards this goal, worst-case corner extraction aims to identify the unique process condition at which a given circuit fails to work [10]-[11]. Once the worst-case corners are determined, designers can simulate their circuit at these corners, find the reason for its performance failure, and eventually come up with the appropriate solution to improve robustness. From this point of view, worst-case corner extraction enables IC designers to easily debug the circuit so that they can efficiently use their design knowledge for yield enhancement.

Extracting the worst-case corners, however, is not trivial, primarily due to the following two reasons. First, realistic worst-case corners cannot be accurately predicted by IC foundries. Instead, they are topology-dependent and performance-dependent [10]-[11]. An efficient algorithm is required to find the worst-case corner for a particular circuit topology and a particular performance metric. Second but more importantly, quadratic RSM results in a non-convex quadratically constrained quadratic programming (QCQP) problem for corner extraction [10]-[11]. As will be demonstrated by our numerical example in Section 4, directly solving such a non-convex optimization using local search methods (e.g., gradient-based search) can easily get stuck at a local minimum. The challenge here is how to find the worst-case corners both *efficiently* and *robustly*.

In this paper, we propose a new mathematical formulation that converts the non-convex QCQP problem to a convex semi-definite programming (SDP) problem that is easier to solve. The proposed approach is derived from the Lagrange duality theory of nonlinear optimization [15]. It explores the unique property that the QCQP formulated for worst-case corner extraction only contains a single quadratic constraint. In such a special case, the dual form of QCQP is a convex SDP. In addition, under some general assumptions, there is no duality gap between the primal problem (i.e., QCQP) and the dual problem (i.e., SDP). Namely, once the dual problem is solved by convex SDP, the solution of the primal problem can be easily determined. By converting the non-convex QCQP to a convex SDP, we can efficiently and robustly find the worst-case corners with global convergence.

The remainder of this paper is organized as follows. In Section 2, we review the background on principal component analysis, response surface modeling, and performance distribution estimation. We propose our mathematical formulation for worst-case corner extraction in Section 3. The efficacy of the proposed corner extraction is demonstrated by the numerical example in Section 4. Finally, we conclude in Section 5.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC'09, July 26-31, 2009, San Francisco, California, USA
Copyright 2009 ACM 978-1-60558-497-3/09/07....10.00

2. BACKGROUND

2.1 Principal Component Analysis

Principal component analysis (PCA) [13] is a statistical method that finds a set of independent factors to represent a set of correlated Normal random variables. Given N process parameters $X = [x_1 \ x_2 \ \dots \ x_N]^T$, the process variations $\Delta X = X - X_0$, where X_0 contains the mean values of X , is often modeled by multiple zero-mean Normal distributions [2]-[11], and the correlation of ΔX can be represented by a symmetric, positive semi-definite covariance matrix R [13]. PCA decomposes R as [13]:

$$R = V \cdot \Sigma \cdot V^T \quad (1)$$

where $\Sigma = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$ contains the eigenvalues of R , and $V = [V_1 \ V_2 \ \dots \ V_N]$ contains the corresponding eigenvectors that are orthonormal, i.e., $V^T V = I$ (I is the identity matrix). Based on Σ and V , PCA defines a set of new random variables:

$$\Delta Y = \Sigma^{-0.5} \cdot V^T \cdot \Delta X \quad (2)$$

These new random variables in ΔY are called the principal components or factors. It is easy to verify that all principal components in ΔY are independent and standard Normal (i.e., zero mean and unit variance).

2.2 Response Surface Modeling

Given a circuit design, the circuit performance (e.g., delay, gain, etc.) is a function of process parameters (e.g., V_{TH} , T_{OX} , etc.). A circuit performance f can be approximated as a quadratic response surface model (RSM) of the process variations [2]-[11]:

$$f(\Delta Y) = \Delta Y^T \cdot A \cdot \Delta Y + B^T \cdot \Delta Y + C \quad (3)$$

where $\Delta Y = [\Delta y_1 \ \Delta y_2 \ \dots \ \Delta y_N]^T$ represents the principal components extracted by PCA, and $C \in R$ and $B \in R^N$ and $A \in R^{N \times N}$ stand for the model coefficients. The quadratic model in (3) provides superior accuracy over a simple linear model, when applied to capture the large-scale manufacturing variations observed in today's IC technologies.

2.3 Performance Distribution Estimation

Once the quadratic response surface model in (3) is available, it can be used to predict the probability distribution of the given performance metric. Although no analytical form exists to represent the probability distribution of the performance $f(\Delta Y)$, the probability density function (PDF) and the cumulative distribution function (CDF) can be numerically calculated by a number of efficient algorithms, e.g., APEX [9]. The performance distribution can be further used to estimate a number of robustness metrics (e.g., worst-case performance, parametric yield, etc.) of the design. However, knowing the worst-case performance and/or parametric yield only is not sufficient. Circuit designers are particularly interested in the unique process condition (i.e., process corner) at which their circuit fails to work. If the worst-case corners are identified, circuit designers can simulate their circuit at these corners and find the appropriate solution to improve its robustness. Motivated by this observation, we aim to develop an efficient algorithm to find worst-case corners in this paper.

3. EFFICIENT CORNER EXTRACTION

In this section, we describe the proposed corner extraction algorithm in detail and highlight its novelties. We first mathematically formulate the corner extraction problem and then develop a convex semi-definite programming (SDP) method to solve it.

3.1 Mathematical Formulation

Once the response surface model $f(\Delta Y)$ is extracted for a performance metric f , it can be used to estimate the probability density function $pdf(f)$, the cumulative distribution function $cdf(f)$, and finally the worst-case performance f_{WC} . Here, the worst-case performance f_{WC} is defined by a given percentile point of $cdf(f)$. Taking the delay of a digital path as an example, the worst-case delay can be defined by the 99% point of $cdf(f)$ [9].

Our goal for worst-case corner extraction is to find the unique process condition (i.e., the value of ΔY^*) at which $f(\Delta Y^*)$ is equal to the worst-case performance value f_{WC} , i.e.:

$$f(\Delta Y^*) = f_{WC} \quad (4)$$

This problem, however, is not mathematically well-defined. Studying (4), we would notice that there are N problem unknowns (i.e., $\Delta Y^* \in R^N$) but only one equation. In other words, Eq. (4) is underdetermined. Mathematically, we can find an infinite number of solutions (i.e., process corners) that satisfy (4).

On the other hand, not all these process corners are useful from the viewpoint of a circuit designer. Some of these corners will "never" occur, because the probability density function $pdf(f)$ is almost zero at these locations. To find the useful process corners, we must take the probability into account. In particular, we want to find the process corner that is most likely to occur, i.e., the maximum likelihood solution of (4) [10].

Recall that the random variables in ΔY are independent and standard Normal after PCA. To find the maximum likelihood solution of (4), we formulate the following optimization problem:

$$\begin{aligned} \max_{\Delta Y} \quad & \left(\frac{1}{\sqrt{2\pi}} \right)^N \cdot \exp\left(-\|\Delta Y\|_2^2/2\right) \\ \text{S.T.} \quad & f(\Delta Y) = f_{WC} \end{aligned} \quad (5)$$

where $\|\bullet\|_2$ denotes the L_2 -norm of a vector. Since $\exp(\bullet)$ monotonically increases, Eq. (5) can be re-written as:

$$\begin{aligned} \min_{\Delta Y} \quad & \|\Delta Y\|_2^2 \\ \text{S.T.} \quad & f(\Delta Y) = f_{WC} \end{aligned} \quad (6)$$

Substituting (3) into (6) yields the following quadratically constrained quadratic programming (QCQP) problem:

$$\begin{aligned} \min_{\Delta Y} \quad & \|\Delta Y\|_2^2 \\ \text{S.T.} \quad & \Delta Y^T \cdot A \cdot \Delta Y + B^T \cdot \Delta Y + C = f_{WC} \end{aligned} \quad (7)$$

While the cost function in (7) is convex, the constraint set is not convex. In this case, directly solving (7) using local search methods (e.g., gradient-based search) can easily get stuck at a local minimum. In what follows, we will propose a novel technique to convert the non-convex QCQP problem in (7) to a convex semi-definite programming (SDP) problem that is easier to solve.

3.2 Lagrange Dual Formulation

Our proposed SDP formulation is derived via three steps: (a) relax the equality constraint in (7) to an inequality constraint; (b) find the Lagrange dual formulation that turns out to be an SDP problem; (c) demonstrate the strong duality so that solving the dual problem yields the solution of the primal problem. In this sub-section, we describe the mathematical details of all these three steps.

First, we relax the equality constraint in (7) to an inequality constraint that depends on the definition of the worst-case performance. If the worst case is defined as the upper bound of the performance variation (e.g., the worst-case delay of a digital

path), we write the constraint as:

$$\Delta Y^T \cdot A \cdot \Delta Y + B^T \cdot \Delta Y + C \geq f_{WC}. \quad (8)$$

Otherwise, if the worst case is defined as the lower bound of the performance variation (e.g., the worst-case bandwidth of an analog amplifier), the constraint is represented as:

$$\Delta Y^T \cdot A \cdot \Delta Y + B^T \cdot \Delta Y + C \leq f_{WC}. \quad (9)$$

Without loss of generality, we represent both (8) and (9) by the following standard form:

$$\Delta Y^T \cdot \tilde{A} \cdot \Delta Y + \tilde{B}^T \cdot \Delta Y + \tilde{C} \leq 0. \quad (10)$$

For example, to convert (8) to (10), we simply have: $\tilde{A} = -A$, $\tilde{B} = -B$ and $\tilde{C} = f_{WC} - C$.

Based on (10), we obtain the following relaxed formulation of the QCQP problem in (7):

$$\begin{aligned} \min_{\Delta Y} \quad & \|\Delta Y\|_2^2 \\ \text{S.T.} \quad & \Delta Y^T \cdot \tilde{A} \cdot \Delta Y + \tilde{B}^T \cdot \Delta Y + \tilde{C} \leq 0 \end{aligned} \quad (11)$$

It is important to note that the two optimization problems in (7) and (11) are exactly equivalent for our corner extraction application. Namely, solving (11) results in the solution of (7). Intuitively, if the constraint in (11) is not active, removing the constraint does not change its solution. In this case, the optimization becomes unconstrained and its solution is simply $\Delta Y = 0$. This, however, should never happen in our application, since the worst-case corner cannot be at the nominal condition. For this reason, the constraint in (11) must be active. Mathematically, this conclusion can be formally proven by using the Karush-Kuhn-Tucker condition from the optimization theory [15].

The relaxed optimization in (11), however, may not be convex either. As will be demonstrated by the numerical example in Section 4, the quadratic coefficient matrix \tilde{A} in (11) can be neither positive semi-definite nor negative semi-definite. In this case, the constraint set in (11) is not convex. To efficiently solve (11), we write the corresponding Lagrange dual problem [15]:

$$\begin{aligned} \max_{\lambda, \gamma} \quad & \gamma \\ \text{S.T.} \quad & \lambda \geq 0 \\ & \begin{bmatrix} I + \lambda \tilde{A} & 0.5 \tilde{B} \\ 0.5 \tilde{B}^T & \lambda \tilde{C} - \gamma \end{bmatrix} \geq 0 \end{aligned} \quad (12)$$

Eq. (12) is a semi-definite programming (SDP) problem [15]. It can be proven that SDP is convex and, hence, it can be solved both robustly and efficiently [15].

While the dual problem in (12) is convex and easy to solve, we need to further demonstrate that solving the dual problem in (12) yields the solution of the primal problem in (11). Denote the cost function value at the optimal solution as p^* and d^* for (11) and (12), respectively. The Lagrange duality theorem guarantees [15]:

$$p^* \geq d^*. \quad (13)$$

Namely, d^* is a lower bound of p^* . In addition, the following theorem gives a sufficient condition for strong duality [15].

Theorem 1: Strong duality holds for (11) and (12), i.e., $p^* = d^*$, if there exists an ΔY with:

$$\Delta Y^T \cdot \tilde{A} \cdot \Delta Y + \tilde{B}^T \cdot \Delta Y + \tilde{C} < 0. \quad (14)$$

In other words, the constraint set in (11) is strictly feasible.

Eq. (14) is referred to as the Slater's constraint qualification. It is obvious that this condition is typically satisfied for our corner extraction application. Namely, we can find a process corner ΔY at which the performance $f(\Delta Y)$ is worse than f_{WC} .

Solving the SDP in (12) yields the optimal value $\lambda = \lambda^*$ that maximizes the cost function γ . Once λ^* is known, the solution ΔY of (11) is given by [15]:

$$\Delta Y^* = -\frac{\lambda^*}{2} \cdot (I + \lambda^* \cdot \tilde{A})^{-1} \cdot \tilde{B}. \quad (15)$$

Eq. (15) gives the unique process corner where the performance $f(\Delta Y)$ reaches the worst-case value f_{WC} .

3.3 Summary

Algorithm 1: Worst-case corner extraction

1. Given a performance function $f(\Delta Y)$ of interest, generate a number of sampling points $\{(\Delta Y_{(i)}, f_{(i)}); i = 1, 2, \dots, K\}$ based on the design of experiments (DOE) [14].
2. Fit the quadratic performance model in (3) to approximate $f(\Delta Y)$.
3. Based on the fitted quadratic performance model, predict the cumulative distribution function $cdf(f)$ by either Monte Carlo analysis or APEX [9].
4. Estimate the worst-case performance value f_{WC} that is defined by the 99% (or 1%) point of $cdf(f)$.
5. Based on the quadratic model coefficients A , B and C in (3) and the worst-case performance value f_{WC} , calculate the coefficients \tilde{A} , \tilde{B} and \tilde{C} in the standard form (10) by using (8)-(9).
6. Solve the semi-definite programming (SDP) in (12) for λ^* .
7. Calculate the worst-case corner ΔY^* using (15).

Algorithm 1 summarizes the major steps of the proposed worst-case corner extraction: (a) quadratic performance modeling, (b) worst-case performance estimation, and (c) worst-case corner extraction. Once the corners are extracted, designers can simulate their circuit at these corners, find the reason for its performance failure, and eventually come up with the appropriate solution to improve robustness.

4. NUMERICAL EXAMPLE

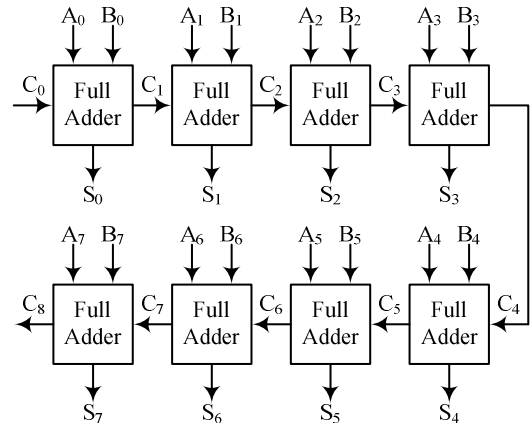


Fig 1. Simplified circuit schematic of an 8-bit ripple carry adder.

In this section, we demonstrate the efficacy of the proposed corner extraction algorithm using an 8-bit ripple carry adder, as shown in Fig 1. The circuit is implemented in a commercial CMOS process. It consists of a chain of 1-bit adders. In this example, only inter-die variations are considered for MOS transistors. After PCA based on foundry data, 34 independent random variables are extracted to model these variations. It should

be noted, however, that nothing would preclude us from handling intra-die variations by using the proposed methodology.

In this example, we are interested in the variability of the propagation delay from “B₀” to “S₇”. We first apply quadratic performance modeling [14] to approximate the delay (i.e., f) as a function of the random variations (i.e., ΔY). The relative modeling error is 0.39%. Fig 2 shows the eigenvalues of the quadratic coefficient matrix A for the performance model $f(\Delta Y)$. Similar to the previous example, the quadratic function $f(\Delta Y)$ is neither positive semi-definite nor negative semi-definite in this example.

Next, we apply Monte Carlo analysis to the quadratic performance model $f(\Delta Y)$, and calculate the worst-case delay that is defined as the 99% point of $cdf(f)$. The semi-definite programming (SDP) problem in (12) is then solved by CVX [12], and the worst-case corner ΔY^* is calculated by using (15). In this example, the runtime for SDP is 1.5 seconds on a LINUX 2.8GHz server with 2GB memory.

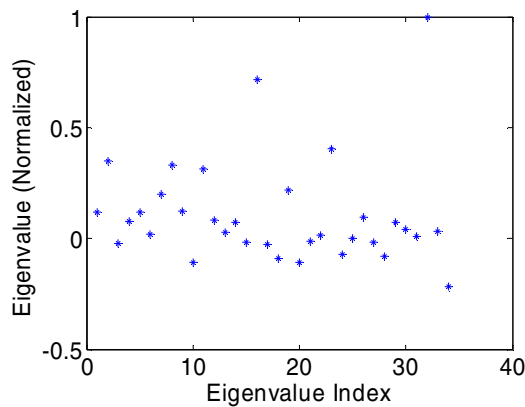


Fig 2. Eigenvalues of the quadratic coefficient matrix A for the performance model $f(\Delta Y)$.

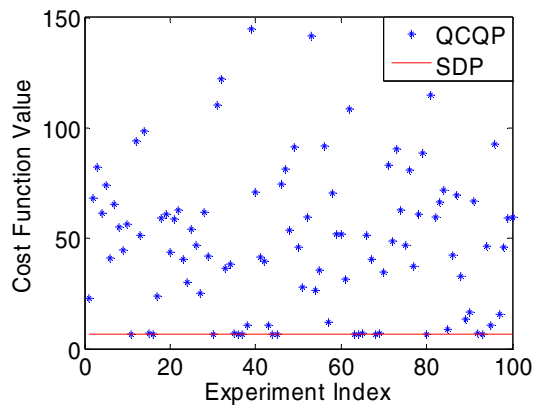


Fig 3. The minimized cost function $\|\Delta Y\|_2^2$ by semi-definite programming (SDP) and 100 runs of quadratically constrained quadratic programming (QCQP) with random initial guess.

For testing and comparison, we apply sequential quadratic programming (SQP) [15] to solve the quadratically constrained quadratic programming (QCQP) problem in (11) to extract the corner ΔY^* . One hundred independent QCQP runs are performed with random initial guess in order to check the convergence property. As shown in Fig 3, a large number of (83 out of 100) QCQP runs get stuck at a local minimum. It, in turn, demonstrates that QCQP fails to robustly extract the worst-case corner ΔY^*

from a non-convex quadratic model.

5. CONCLUSIONS

In this paper, we propose a novel semi-definite programming (SDP) method to efficiently extract design-specific worst-case corners for nanoscale circuits. The proposed SDP approach is facilitated by the Lagrange duality theory of nonlinear optimization [15]. It converts a non-convex quadratically constrained quadratic programming (QCQP) problem to a convex SDP problem that is easier to solve. Our numerical experiments demonstrate that while the traditional QCQP fails to achieve global convergence, our proposed SDP formulation can find the worst-case corners both robustly and efficiently. The proposed SDP method can be further incorporated into a statistical circuit analysis flow to provide design insights and assist failure diagnosis.

6. REFERENCES

- [1] Semiconductor Industry Associate, *International Technology Roadmap for Semiconductors*, 2007.
- [2] X. Li, J. Le, L. Pileggi and A. Strojwas, “Projection-based performance modeling for inter/intra-die variations,” *IEEE ICCAD*, pp. 721-727, 2005.
- [3] Z. Feng and P. Li, “Performance-oriented statistical parameter reduction of parameterized systems via reduced rank regression,” *IEEE ICCAD*, pp. 868-875, 2006.
- [4] A. Singhee and R. Rutenbar, “Beyond low-order statistical response surfaces: latent variable regression for efficient, highly nonlinear fitting,” *IEEE DAC*, pp. 256-261, 2007.
- [5] A. Mitev, M. Marefat, D. Ma and J. Wang, “Principle Hessian direction based parameter reduction for interconnect networks with process variation,” *IEEE ICCAD*, pp. 632-637, 2007.
- [6] X. Li and H. Liu, “Statistical regression for efficient high-dimensional modeling of analog and mixed-signal performance variations,” *IEEE DAC*, pp. 38-43, 2008.
- [7] G. Debyser and G. Gielen, “Efficient analog circuit synthesis with simultaneous yield and robustness optimization,” *IEEE ICCAD*, pp. 308-311, 1998.
- [8] F. Schenkel, M. Pronath, S. Zizala, R. Schwencker, H. Graeb and K. Antreich, “Mismatch analysis and direct yield optimization by spec-wise linearization and feasibility-guided search,” *IEEE DAC*, pp. 858-863, 2001.
- [9] X. Li, J. Le, P. Gopalakrishnan and L. Pileggi, “Asymptotic probability extraction for non-normal distributions of circuit performance,” *IEEE ICCAD*, pp. 2-9, 2004.
- [10] A. Dharchoudhury and S. Kang, “Worst-case analysis and optimization of VLSI circuit performances,” *IEEE Trans. CAD*, vol. 14, no. 4, pp. 481-492, Apr. 1995.
- [11] M. Sengupta, S. Saxena, L. Daldoss, G. Kramer, S. Minehane and J. Cheng, “Application-specific worst case corners using response surfaces and statistical models,” *IEEE Trans. CAD*, vol. 24, no. 9, pp. 1372-1380, 2005.
- [12] M. Grant and S. Boyd, *CVX Users’ Guide*, 2008. (www.stanford.edu/~boyd/cvx/)
- [13] G. Seber, *Multivariate Observations*, Wiley Series, 1984.
- [14] R. Myers and D. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley-Interscience, 2002.
- [15] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.