

Efficient Parametric Yield Estimation of Analog/Mixed-Signal Circuits via Bayesian Model Fusion

Xin Li¹, Wangyang Zhang¹, Fa Wang¹, Shupeng Sun¹ and Chenjie Gu²

¹Carnegie Mellon University, Pittsburgh, PA 15213, {xinli, wangyan1, fwang1, shupengs}@andrew.cmu.edu

²Intel Strategic CAD Labs, Hillsboro, OR 97124, chenjie.gu@intel.com

ABSTRACT

Parametric yield estimation is one of the most critical-yet-challenging tasks for designing and verifying nanoscale analog and mixed-signal circuits. In this paper, we propose a novel Bayesian model fusion (BMF) technique for efficient parametric yield estimation. Our key idea is to borrow the simulation data from an early stage (e.g., schematic-level simulation) to efficiently estimate the performance distributions at a late stage (e.g., post-layout simulation). BMF statistically models the correlation between early-stage and late-stage performance distributions by Bayesian inference. In addition, a convex optimization is formulated to solve the unknown late-stage performance distributions both accurately and robustly. Several circuit examples designed in a commercial 32 nm CMOS process demonstrate that the proposed BMF technique achieves up to 3.75× runtime speedup over the traditional kernel estimation method.

1. INTRODUCTION

The aggressive scaling of integrated circuit (IC) technologies has brought about large-scale process variations that substantially impact the performance metrics of nanoscale analog and mixed-signal (AMS) circuits [1]-[3]. Parametric yield becomes an important metric that must be carefully optimized by AMS designers in order to produce reliable silicon chips after the manufacturing process. For this reason, accurately modeling and estimating parametric yield of AMS circuits has been considered as a critical task within the AMS design flow. Once the parametric yield of a given AMS circuit is known, it can be further used to guide circuit optimization for yield enhancement [4]-[10].

During the past two decades, a large number of techniques have been proposed for parametric yield estimation. These traditional approaches can be classified into two broad categories: (i) Monte Carlo approaches [11], [16], [20], and (ii) performance modeling approaches [12]-[13], [17]. The Monte Carlo methods first generate a number of random samples based on the statistics of process variations. Next, they estimate the probability distribution of the performance of interest (e.g., delay, gain, etc) based on these random samples. On the other hand, the performance modeling approaches first approximate the performance of interest as an analytical (i.e., either linear or nonlinear) function of device-level variations (e.g., ΔV_{TH} , ΔT_{OX} , etc). Once such a performance model is available, the performance distribution can be estimated by several numerical

Permission to make digital or hard copies of all or part 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.
IEEE/ACM International Conference on Computer-Aided Design (ICCAD) 2012, November 5-8, 2012, San Jose, California, USA
Copyright © 2012 ACM 978-1-4503-1573-9/12/11... \$15.00

algorithms such as moment matching [12].

These traditional techniques for parametric yield estimation have been successfully applied to a large number of practical applications. However, with the continuous scaling of IC technologies, today's AMS circuits become increasingly complex, as circuit designers have adopted a number of *non-traditional* methodologies (e.g., multi-mode operation, adaptive self-healing, etc) to address the design challenges associated with technology scaling (e.g., reduced voltage headroom, increased process variation, etc). These recent trends of AMS circuits have brought up new challenges and opportunities for parametric yield estimation and suggested an immediate need to re-visit this area.

On one hand, simulating today's complex AMS circuits is extremely expensive due to their large sizes. For an industrial AMS design such as phase-locked loop or high-speed I/O link, one single transistor-level simulation may take a few days or even a few weeks to finish. In this case, it would be computationally inefficient, if not impossible, to apply the traditional yield estimation techniques, as they often require collecting 100~1000 simulation samples to accurately estimate the performance distribution of interest. The open question here is how to develop new algorithms and methodologies to appropriately address this *complexity* issue.

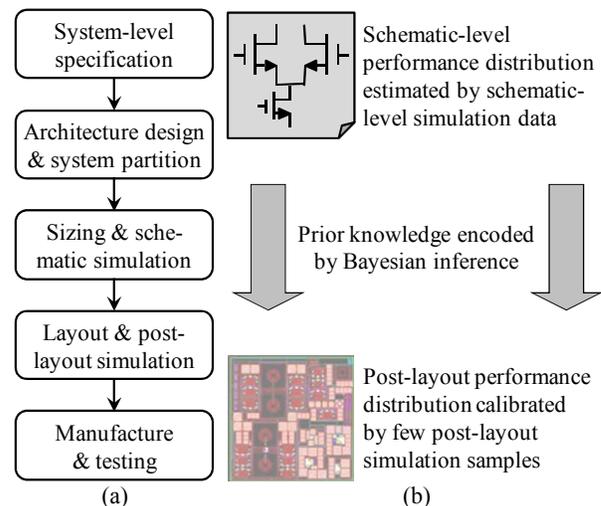


Figure 1. (a) Today's AMS design flow typically involves multiple stages. (b) A specific application example of Bayesian model fusion (BMF) is shown to borrow schematic-level simulation data to efficiently estimate post-layout performance distribution.

On the other hand, in order to reliably design complex AMS circuits, it becomes necessary for circuit designers to follow a hierarchical design flow that involves multiple stages, as shown in Figure 1(a). At each stage, the circuit is simulated to create simulation data to verify all performance metrics, before moving to the next design stage. Since these simulation data collected

from multiple design stages come from the same circuit, they are expected to be strongly correlated. Hence, it is possible to *borrow* the simulation samples from an early stage to reduce the number of required simulation runs for parametric yield estimation at a late stage. In other words, the multi-stage AMS design flow creates a completely new opportunity that has not been explored by the traditional yield estimation methods.

In this paper, we propose a novel technique, referred to as *Bayesian model fusion* (BMF), to estimate the AMS performance distribution by using a small number of simulation samples. Starting from the early-stage performance distribution that is estimated from a set of early-stage simulation samples, BMF first extracts a model template to capture the performance distribution and statistically encodes it as our *prior* knowledge. Next, the aforementioned model template is further calibrated by applying Bayesian inference to very few late-stage simulation samples to accurately estimate the late-stage performance distribution. From this point of view, by *fusing* the early-stage and late-stage performance distributions through Bayesian inference, the computational cost for late-stage parametric yield estimation can be substantially reduced. Figure 1(b) shows a specific application example of our proposed BMF method where the schematic-level simulation data are used to help to estimate the post-layout performance distribution. As will be demonstrated by several circuit examples in Section 5, BMF achieves up to 3.75× runtime speedup compared to the traditional kernel estimation technique.

The remainder of this paper is organized as follows. In Section 2 we develop the mathematical formulation for the proposed distribution estimation problem, and then derive the BMF algorithm in Section 3. Several implementation details are further discussed in Section 4. The efficiency of BMF is demonstrated by several circuit examples in Section 5. Finally, we conclude in Section 6.

2. MATHEMATICAL FORMULATION

Without loss of generality, we consider a given performance of interest g (e.g., delay of a digital buffer, gain of an analog amplifier, etc). Due to process variations, the value of g is not deterministic. Instead, it must be statistically modeled as a random variable that is described by its probability density function $pdf(g)$. In this paper, we use different symbols $pdf_E(g)$ and $pdf_L(g)$ to distinguish the early-stage (e.g., schematic-level) and late-stage (e.g., post-layout) performance distributions. Our goal is to accurately and efficiently estimate the late-stage performance distribution $pdf_L(g)$ based on the early-stage distribution $pdf_E(g)$ and a small number of late-stage simulation samples. To this end, we approximate the one-dimensional probability density functions $pdf_E(g)$ and $pdf_L(g)$ as the linear combinations of a set of basis functions

$$pdf_E(g) = \sum_{n=1}^N \alpha_{E,n} \cdot b_n(g) \quad (1)$$

$$pdf_L(g) = \sum_{n=1}^N \alpha_{L,n} \cdot b_n(g) \quad (2)$$

where $\{b_n(g); n = 1, 2, \dots, N\}$ contains the basis functions that are applied, $\{\alpha_{E,n}; n = 1, 2, \dots, N\}$ and $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ contain the coefficients associated with the basis functions, and N represents the total number of these basis functions.

There are two important clarifications that should be made regarding the approximations in (1)-(2). First, we assume that the probability density functions $pdf_E(g)$ and $pdf_L(g)$ are defined over

the interval

$$g \in [g_{MIN}, g_{MAX}]. \quad (3)$$

In general, g_{MIN} and g_{MAX} can be $-\infty$ and $+\infty$, respectively. Namely, the performance value g may not be bounded. However, in most practical applications, we can approximately set $[g_{MIN}, g_{MAX}]$ as a finite interval. For example, consider the case where g follows a Gaussian distribution and, hence, is unbounded. In this example, we can approximately set $g_{MIN} = \mu_g - \zeta \cdot \sigma_g$ and $g_{MAX} = \mu_g + \zeta \cdot \sigma_g$ where μ_g and σ_g are the mean and standard deviation of the random variable g and ζ is a positive constant. As long as ζ is sufficiently large (e.g., $\zeta = 4-5$), the major portion of the distribution falls inside the interval $[g_{MIN}, g_{MAX}]$ and, hence, the approximation error is negligible.

Second, there are multiple possible choices to select the basis functions $\{b_n(g); n = 1, 2, \dots, N\}$ (e.g., orthogonal polynomials [14], wavelet basis functions [19], Fourier basis functions [15], etc). For our numerical examples in Section 5, we will use DCT (discrete cosine transform) basis functions [19] to model the probability distributions of several circuit performance metrics. However, it should be noted that the BMF framework developed in this paper is generally applicable to other basis functions as well.

Given the probability density functions $pdf_E(g)$ and $pdf_L(g)$ in (1)-(2) over the interval $[g_{MIN}, g_{MAX}]$, the cumulative density functions $cdf_E(g)$ and $cdf_L(g)$ are simply equal to their integrals

$$cdf_E(g) = \int_{g_{MIN}}^g pdf_E(x) \cdot dx = \sum_{n=1}^N \alpha_{E,n} \cdot B_n(g) \quad (4)$$

$$cdf_L(g) = \int_{g_{MIN}}^g pdf_L(x) \cdot dx = \sum_{n=1}^N \alpha_{L,n} \cdot B_n(g) \quad (5)$$

where $B_n(g)$ denotes the integral of $b_n(g)$

$$B_n(g) = \int_{g_{MIN}}^g b_n(x) \cdot dx \quad (n=1,2,\dots,N). \quad (6)$$

For a given set of basis functions $\{b_n(g); n = 1, 2, \dots, N\}$, it is straightforward to verify that the performance distributions $pdf_E(g)$ and $pdf_L(g)$ are uniquely determined by the coefficients $\{\alpha_{E,n}; n = 1, 2, \dots, N\}$ and $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$. Hence, we need to accurately solve the late-stage coefficients $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ in order to estimate the late-stage performance distribution $pdf_L(g)$.

Unlike the traditional yield estimation techniques that completely ignore the correlation between the early-stage performance distribution $pdf_E(g)$ and the late-stage performance distribution $pdf_L(g)$, our proposed BMF method fully exploits such correlation information to reduce the number of required simulation samples at the late stage. As such, the computational cost of late-stage yield estimation can be substantially reduced. In the following section, we will derive the proposed BMF algorithm in detail.

3. BAYESIAN MODEL FUSION

Our proposed BMF framework contains two core components: (i) statistically encoding the prior knowledge learned from the early-stage performance distribution $pdf_E(g)$, and (ii) applying maximum-a-posteriori (MAP) estimation to determine the late-stage performance distribution $pdf_L(g)$ through Bayesian inference. In this section, we describe the mathematical details of the BMF algorithm and highlight its novelty.

3.1 Prior Knowledge Definition

The key idea of BMF is to borrow the early-stage performance distribution $pdf_E(g)$ to define our prior knowledge for the late-stage performance distribution $pdf_L(g)$. On one hand, since $pdf_E(g)$ and $pdf_L(g)$ represent the probability distributions of the same performance metric for the same circuit, we expect that these distribution functions are similar. In other words, the difference between the coefficients $\{\alpha_{E,n}; n = 1, 2, \dots, N\}$ and $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ is small. On the other hand, the late-stage distribution $pdf_L(g)$ is not exactly identical to the early-stage distribution $pdf_E(g)$. Note that $pdf_E(g)$ and $pdf_L(g)$ may be different, because the late-stage simulation often involves more detailed circuit models than the early-stage simulation. For instance, the post-layout simulation of an AMS circuit includes the device and interconnect parasitics that are not available during the schematic-level simulation of the same circuit.

To account for the uncertainties associated with the late-stage coefficients $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$, we statistically model them as random variables with the following distributions

$$pdf(\alpha_{L,n}) = \frac{1}{\sqrt{2\pi} \cdot \lambda \cdot |\alpha_{E,n}|} \cdot \exp\left[-\frac{(\alpha_{L,n} - \alpha_{E,n})^2}{2 \cdot \lambda^2 \cdot \alpha_{E,n}^2}\right] \quad (7)$$

$(n = 1, 2, \dots, N)$

where $pdf(\alpha_{L,n})$ represents a Gaussian distribution with the mean value $\alpha_{E,n}$ and the standard deviation $\lambda \cdot |\alpha_{E,n}|$. In (7), λ denotes a positive parameter that controls the variance of the distribution. The appropriate value of λ can be determined by the cross validation method discussed in Section 4.2.

The probability density function $pdf(\alpha_{L,n})$ in (7) is referred to as the *prior distribution* [20] for the late-stage coefficient $\alpha_{L,n}$. It has a two-fold meaning. First, the Gaussian distribution $pdf(\alpha_{L,n})$ is maximized at its mean value $\alpha_{L,n} = \alpha_{E,n}$, implying that the early-stage coefficient $\alpha_{E,n}$ and the late-stage coefficient $\alpha_{L,n}$ are likely to be similar. In other words, since the Gaussian distribution $pdf(\alpha_{L,n})$ exponentially decays with $(\alpha_{L,n} - \alpha_{E,n})^2$, it is unlikely to observe a late-stage coefficient $\alpha_{L,n}$ that is extremely different from the early-stage coefficient $\alpha_{E,n}$.

Second, the standard deviation of the prior distribution $pdf(\alpha_{L,n})$ is proportional to $|\alpha_{E,n}|$. It means that the absolute difference between the late-stage coefficient $\alpha_{L,n}$ and the early-stage coefficient $\alpha_{E,n}$ can be large (or small), if the magnitude of the coefficient $\alpha_{E,n}$ is large (or small). Restating in words, each late-stage coefficient $\alpha_{L,n}$ has been provided with a relatively equal opportunity to differ from the corresponding early-stage coefficient $\alpha_{E,n}$.

To completely define the prior distribution, we further assume that the late-stage coefficients $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ are mutually independent. Hence, their joint probability density function is represented as

$$pdf(\mathbf{\alpha}_L) = \frac{\exp\left[-\frac{1}{2} \cdot (\mathbf{\alpha}_L - \boldsymbol{\mu}_{\alpha L})^T \cdot \boldsymbol{\Sigma}_{\alpha L}^{-1} \cdot (\mathbf{\alpha}_L - \boldsymbol{\mu}_{\alpha L})\right]}{(\sqrt{2\pi})^N \cdot \sqrt{\det(\boldsymbol{\Sigma}_{\alpha L})}} \quad (8)$$

where $\det(\bullet)$ denotes the determinant of a matrix and $\mathbf{\alpha}_L \in R^N$ is a vector containing all late-stage coefficients

$$\mathbf{\alpha}_L = [\alpha_{L,1} \quad \alpha_{L,2} \quad \dots \quad \alpha_{L,N}]^T \quad (9)$$

The mean vector $\boldsymbol{\mu}_{\alpha L} \in R^N$ and the covariance matrix $\boldsymbol{\Sigma}_{\alpha L} \in R^{N \times N}$ are

$$\boldsymbol{\mu}_{\alpha L} = [\alpha_{E,1} \quad \alpha_{E,2} \quad \dots \quad \alpha_{E,N}]^T \quad (10)$$

$$\boldsymbol{\Sigma}_{\alpha L} = \lambda^2 \cdot \text{diag}(\alpha_{E,1}^2, \alpha_{E,2}^2, \dots, \alpha_{E,N}^2) \quad (11)$$

where $\text{diag}(\bullet)$ represents a diagonal matrix. The independence assumption in (8)-(11) simply implies that we do not know the correlation information between the late-stage coefficients $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ as a prior. However, the correlation between $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ will be taken into account by our proposed BMF algorithm through two different avenues: (i) when we legalize the prior distribution in (8) to guarantee that $pdf_L(g)$ in (2) is a valid probability density function, and (ii) when we calculate the posterior distribution of $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ once the late-stage simulation samples are available. In what follows, we will discuss these two topics (i.e., prior distribution legalization and posterior distribution calculation) in Section 3.2 and 3.3 respectively.

3.2 Prior Distribution Legalization

The purpose of prior distribution legalization is to add additional constraints to appropriately model the uncertainties for the late-stage coefficients $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$. Remember that the prior distribution in (8) assumes that all coefficients $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ are mutually independent and each coefficient follows a Gaussian distribution. Since these Gaussian distributions associated with $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ are not bounded, our definition of prior distribution in (8) allows a coefficient $\alpha_{L,n}$ to take any value ranging from $-\infty$ to $+\infty$. However, a large number of these values are not valid, because the probability density function $pdf_L(g)$, determined by the coefficients $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ in (2), must satisfy the following two properties.

Property 1: For a probability density function $pdf_L(g)$ defined over the interval $[g_{MIN}, g_{MAX}]$, the following integral must equal 1

$$\int_{g_{MIN}}^{g_{MAX}} pdf_L(x) \cdot dx = cdf_L(g_{MAX}) = \sum_{n=1}^N \alpha_{L,n} \cdot B_n(g_{MAX}) = 1. \quad (12)$$

Property 2: For a probability density function $pdf_L(g)$ defined over the interval $[g_{MIN}, g_{MAX}]$, it must be non-negative at any $g \in [g_{MIN}, g_{MAX}]$

$$\forall g \in [g_{MIN}, g_{MAX}]: pdf_L(g) \geq 0. \quad (13)$$

Eq. (12) requires that the linear combination of all late-stage coefficients $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ must be equal to the constant 1. It, in turn, implies that the coefficients $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ cannot be considered as independent random variables. Instead, we must appropriately model the correlation information between $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ such that Eq. (12) holds.

Towards this goal, we define a set of new coefficients $\{\beta_{L,n}; n = 1, 2, \dots, N\}$ by the following linear transformation

$$\boldsymbol{\beta}_L = \mathbf{P}^T \cdot \boldsymbol{\alpha}_L \quad (14)$$

where $\boldsymbol{\beta}_L \in R^N$ is a column vector containing all variables $\{\beta_{L,n}; n = 1, 2, \dots, N\}$

$$\boldsymbol{\beta}_L = [\beta_{L,1} \quad \beta_{L,2} \quad \dots \quad \beta_{L,N}]^T \quad (15)$$

and $\mathbf{P} \in R^{N \times N}$ is the transformation matrix containing N column vectors $\{\mathbf{P}_n; n = 1, 2, \dots, N\}$

$$\mathbf{P} = [\mathbf{P}_1 \quad \mathbf{P}_2 \quad \dots \quad \mathbf{P}_N]. \quad (16)$$

Based on (8) and (14), it is easy to verify that the new coefficients $\{\beta_{L,n}; n = 1, 2, \dots, N\}$ follow a joint Gaussian distribution [16]

$$pdf(\boldsymbol{\beta}_L) = \frac{\exp\left[-\frac{1}{2} \cdot (\boldsymbol{\beta}_L - \boldsymbol{\mu}_{\beta L})^T \cdot \boldsymbol{\Sigma}_{\beta L}^{-1} \cdot (\boldsymbol{\beta}_L - \boldsymbol{\mu}_{\beta L})\right]}{(\sqrt{2\pi})^N \cdot \sqrt{\det(\boldsymbol{\Sigma}_{\beta L})}} \quad (17)$$

where the mean vector $\boldsymbol{\mu}_{\beta_L} \in R^N$ and the covariance matrix $\boldsymbol{\Sigma}_{\beta_L} \in R^{N \times N}$ are

$$\boldsymbol{\mu}_{\beta_L} = \mathbf{P}^T \cdot \boldsymbol{\mu}_{\mathbf{a}_L} \quad (18)$$

$$\boldsymbol{\Sigma}_{\beta_L} = \mathbf{P}^T \cdot \boldsymbol{\Sigma}_{\mathbf{a}_L} \cdot \mathbf{P}. \quad (19)$$

In general, the coefficients $\{\beta_{L,n}; n = 1, 2, \dots, N\}$ are correlated after the linear transformation, as shown by the covariance matrix $\boldsymbol{\Sigma}_{\beta_L}$ in (19).

In this paper, the transformation matrix \mathbf{P} in (16) is selected to meet the following two criteria. First, the N vectors $\{\mathbf{P}_n; n = 1, 2, \dots, N\}$ should be linearly independent so that the matrix \mathbf{P} is invertible

$$\boldsymbol{\alpha}_L = \mathbf{P}^{-T} \cdot \boldsymbol{\beta}_L. \quad (20)$$

Second, we set the N th column vector \mathbf{P}_N as

$$\mathbf{P}_N = [B_1(g_{MAX}) \ B_2(g_{MAX}) \ \dots \ B_N(g_{MAX})]^T. \quad (21)$$

Hence, the equality constraint in (12) becomes

$$\beta_{L,N} = \mathbf{P}_N^T \cdot \boldsymbol{\alpha}_L = 1. \quad (22)$$

Namely, $\beta_{L,N}$ must be equal to 1 in order to guarantee that the probability density function $pdf_i(g)$ is valid.

Eq. (22) leads to a two-fold consequence. First, since the value of $\beta_{L,N}$ is determined, we do not need to consider it as a problem unknown any more. Second, but more importantly, since $\beta_{L,N}$ is correlated to $\{\beta_{L,n}; n = 1, 2, \dots, N-1\}$, knowing $\beta_{L,N} = 1$ will change the probability distribution for $\{\beta_{L,n}; n = 1, 2, \dots, N-1\}$ (i.e., our prior knowledge about these coefficients). The influence of $\beta_{L,N}$ on $\{\beta_{L,n}; n = 1, 2, \dots, N-1\}$ can be mathematically expressed by the following conditional probability density function

$$pdf(\boldsymbol{\beta}_{L,N} | \beta_{L,N}) = \frac{pdf(\boldsymbol{\beta}_{L,N}, \beta_{L,N})}{pdf(\beta_{L,N})} \quad (23)$$

where $\boldsymbol{\beta}_{L,N}$ denotes the vector $\boldsymbol{\beta}_L$ with the N th element $\beta_{L,N}$ removed. Given the joint Gaussian distribution in (17)-(19), it can be proven that the conditional distribution $pdf(\boldsymbol{\beta}_{L,N} | \beta_{L,N})$ remains Gaussian [20]

$$pdf(\boldsymbol{\beta}_{L,N} | \beta_{L,N}) = \frac{1}{(\sqrt{2\pi})^{N-1} \cdot \sqrt{\det(\boldsymbol{\Sigma}_{\beta_{L,N}})}} \exp\left[-\frac{1}{2} \cdot (\boldsymbol{\beta}_{L,N} - \boldsymbol{\mu}_{\beta_{L,N}})^T \cdot \boldsymbol{\Sigma}_{\beta_{L,N}}^{-1} \cdot (\boldsymbol{\beta}_{L,N} - \boldsymbol{\mu}_{\beta_{L,N}})\right] \quad (24)$$

where the mean vector $\boldsymbol{\mu}_{\beta_{L,N}} \in R^{N-1}$ and the covariance matrix $\boldsymbol{\Sigma}_{\beta_{L,N}} \in R^{(N-1) \times (N-1)}$ are [20]

$$\boldsymbol{\mu}_{\beta_{L,N}} = \boldsymbol{\mu}_{\beta_L, \tilde{N}} - \boldsymbol{\Delta}_{\beta_L, \tilde{N}\tilde{N}}^{-1} \cdot \boldsymbol{\Delta}_{\beta_L, \tilde{N}N} \cdot (1 - \mu_{\beta_{L,N}}) \quad (25)$$

$$\boldsymbol{\Sigma}_{\beta_{L,N}} = \boldsymbol{\Delta}_{\beta_L, \tilde{N}\tilde{N}}^{-1}. \quad (26)$$

In (25)-(26), $\boldsymbol{\mu}_{\beta_L, \tilde{N}} \in R^{N-1}$, $\mu_{\beta_{L,N}} \in R$, $\boldsymbol{\Delta}_{\beta_L, \tilde{N}\tilde{N}} \in R^{(N-1) \times (N-1)}$ and $\boldsymbol{\Delta}_{\beta_L, \tilde{N}N} \in R^{(N-1)}$ are defined as the following sub-matrices

$$\boldsymbol{\mu}_{\beta_L} = \begin{bmatrix} \boldsymbol{\mu}_{\beta_L, \tilde{N}} \\ \mu_{\beta_{L,N}} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_{\beta_L}^{-1} = \begin{bmatrix} \boldsymbol{\Delta}_{\beta_L, \tilde{N}\tilde{N}} & \boldsymbol{\Delta}_{\beta_L, \tilde{N}N} \\ \boldsymbol{\Delta}_{\beta_L, \tilde{N}N} & \boldsymbol{\Delta}_{\beta_L, NN} \end{bmatrix}. \quad (27)$$

The conditional probability density function $pdf(\boldsymbol{\beta}_{L,N} | \beta_{L,N})$ in (24)-(27) specifies the prior distribution of the coefficients $\{\beta_{L,n}; n = 1, 2, \dots, N-1\}$ that are guaranteed to satisfy the linear constraint in (12).

On the other hand, we need to further consider the inequality constraint in (13). Given the probability density function in (2), the linear transformation in (14) and the fact $\beta_{L,N} = 1$ shown in (22), it is straightforward to represent the inequality constraint in

(13) as a function of $\{\beta_{L,n}; n = 1, 2, \dots, N-1\}$

$$\forall g \in [g_{MIN}, g_{MAX}]: [b_1(g) \ \dots \ b_N(g)] \cdot \mathbf{P}^{-T} \cdot \begin{bmatrix} \boldsymbol{\beta}_{L,N} \\ 1 \end{bmatrix} \geq 0. \quad (28)$$

Note that Eq. (28) must be satisfied for any $g \in [g_{MIN}, g_{MAX}]$ (i.e., an infinite number of possible values of g). In practice, we can define a set of one-dimensional grid points $\{g_{GRID,m}; m = 1, 2, \dots, M\}$ within the interval $[g_{MIN}, g_{MAX}]$ and then force (28) to hold at these grid points

$$[b_1(g_{GRID,m}) \ \dots \ b_N(g_{GRID,m})] \cdot \mathbf{P}^{-T} \cdot \begin{bmatrix} \boldsymbol{\beta}_{L,N} \\ 1 \end{bmatrix} \geq 0 \quad (29)$$

$(m = 1, 2, \dots, M)$

Equivalently, Eq. (29) can be re-written in the matrix form

$$\mathbf{B}_{GRID} \cdot \mathbf{P}^{-T} \cdot \begin{bmatrix} \boldsymbol{\beta}_{L,N} \\ 1 \end{bmatrix} \geq 0 \quad (30)$$

where

$$\mathbf{B}_{GRID} = \begin{bmatrix} b_1(g_{GRID,1}) & b_2(g_{GRID,1}) & \dots & b_N(g_{GRID,1}) \\ b_1(g_{GRID,2}) & b_2(g_{GRID,2}) & \dots & b_N(g_{GRID,2}) \\ \vdots & \vdots & \vdots & \vdots \\ b_1(g_{GRID,M}) & b_2(g_{GRID,M}) & \dots & b_N(g_{GRID,M}) \end{bmatrix}. \quad (31)$$

As long as the step size of these grid points is sufficiently small, the linear constraint in (30) provides a good approximation of the original constraint in (28).

In summary, based on the linear transformation in (14), Eq. (24) and (30) define the legalized prior distribution for the coefficients $\{\beta_{L,n}; n = 1, 2, \dots, N-1\}$ such that the constraints in (12)-(13) are guaranteed to hold. In the next sub-section, we will further combine the prior distribution with the late-stage simulation samples to optimally estimate the late-stage performance distribution $pdf_i(g)$.

3.3 Maximum-A-Posteriori Estimation

Given our prior knowledge defined in (24) and (30), we need to further collect a number of late-stage random samples $\{g_{RAND,k}; k = 1, 2, \dots, K\}$ to accurately estimate the coefficients $\{\beta_{L,n}; n = 1, 2, \dots, N-1\}$. These samples can be created by running Monte Carlo simulation for a given AMS circuit. Once these simulation samples are available, they can tell us additional information about the late-stage performance distribution and, hence, help us to determine the coefficient values. In this sub-section, we will borrow the Bayesian inference theory [20] from the statistics community to solve the optimal coefficient values.

Based on Bayes' theorem, the uncertainties of the coefficients $\{\beta_{L,n}; n = 1, 2, \dots, N-1\}$ after knowing a set of simulation samples $\{g_{RAND,k}; k = 1, 2, \dots, K\}$ can be mathematically described by the following *posterior distribution* [20]

$$pdf(\boldsymbol{\beta}_{L,N} | \mathbf{g}_{RAND}) \propto pdf(\boldsymbol{\beta}_{L,N} | \beta_{L,N}) \cdot pdf(\mathbf{g}_{RAND} | \boldsymbol{\beta}_{L,N}) \quad (32)$$

where $\mathbf{g}_{RAND} \in R^K$ is a column vector containing all simulation samples

$$\mathbf{g}_{RAND} = [g_{RAND,1} \ g_{RAND,2} \ \dots \ g_{RAND,K}]^T. \quad (33)$$

In (32), the prior distribution $pdf(\boldsymbol{\beta}_{L,N} | \beta_{L,N})$ is defined by (24). The conditional distribution $pdf(\mathbf{g}_{RAND} | \boldsymbol{\beta}_{L,N})$ is called the *likelihood* function [20]. It measures the probability of observing the random samples $\{g_{RAND,k}; k = 1, 2, \dots, K\}$ associated with the late-stage performance distribution $pdf_i(g)$. Since the samples $\{g_{RAND,k}; k = 1, 2, \dots, K\}$ are generated from K independent Monte Carlo runs, the likelihood function can be represented as

$$pdf(\mathbf{g}_{RAND}|\boldsymbol{\beta}_{L\setminus N}) = \prod_{k=1}^K pdf_L(g_{RAND,k}). \quad (34)$$

Substituting (2), (14) and (22) into (34) yields

$$pdf(\mathbf{g}_{RAND}|\boldsymbol{\beta}_{L\setminus N}) = \prod_{k=1}^K \left\{ \mathbf{b}_{RAND,k} \cdot \mathbf{P}^{-T} \cdot \begin{bmatrix} \boldsymbol{\beta}_{L\setminus N} \\ 1 \end{bmatrix} \right\} \quad (35)$$

where $\mathbf{b}_{RAND,k} \in R^N$ is a row vector containing all basis function values at the k th simulation sample $g_{RAND,k}$

$$\mathbf{b}_{RAND,k} = [b_1(g_{RAND,k}) \quad b_2(g_{RAND,k}) \quad \dots \quad b_N(g_{RAND,k})] \quad (k=1,2,\dots,K) \quad (36)$$

As shown in (35), the likelihood function depends on the unknown coefficients $\{\beta_{L,n}; n = 1, 2, \dots, N-1\}$ that we aim to solve. These coefficients $\{\beta_{L,n}; n = 1, 2, \dots, N-1\}$ control the shape of the probability density function $pdf_L(g)$ and, therefore, directly influence the likelihood function in (35).

Even after the simulation samples $\{g_{RAND,k}; k = 1, 2, \dots, K\}$ are available, the coefficients $\{\beta_{L,n}; n = 1, 2, \dots, N-1\}$ are not deterministic. They must be modeled by the probability density function $pdf(\boldsymbol{\beta}_{L\setminus N} | \mathbf{g}_{RAND})$ (i.e., the posterior distribution) in (32). Depending on the shape of the distribution $pdf(\boldsymbol{\beta}_{L\setminus N} | \mathbf{g}_{RAND})$, these coefficients $\{\beta_{L,n}; n = 1, 2, \dots, N-1\}$ do not take all possible values with equal probability. If the posterior distribution $pdf(\boldsymbol{\beta}_{L\setminus N} | \mathbf{g}_{RAND})$ reaches its maximum value at $\{\beta_{L,n}^*; n = 1, 2, \dots, N-1\}$, the coefficient values $\{\beta_{L,n}^*; n = 1, 2, \dots, N-1\}$ are considered as the optimal estimation, since these coefficient values are most likely to occur. Such a method is referred to as *maximum-a-posteriori* (MAP) estimation in the literature [20].

The aforementioned MAP estimation can be mathematically formulated as an optimization problem

$$\begin{aligned} & \underset{\boldsymbol{\beta}_{L\setminus N}}{\text{maximize}} \quad pdf(\boldsymbol{\beta}_{L\setminus N} | \boldsymbol{\beta}_{L\setminus N}) \cdot pdf(\mathbf{g}_{RAND} | \boldsymbol{\beta}_{L\setminus N}) \\ & \text{subject to} \quad \mathbf{B}_{GRID} \cdot \mathbf{P}^{-T} \cdot \begin{bmatrix} \boldsymbol{\beta}_{L\setminus N} \\ 1 \end{bmatrix} \geq 0 \end{aligned} \quad (37)$$

In (37), the constraint is added to guarantee that the resulting coefficients are legalized, as shown in (30). Substituting (24) and (35) into (37) and taking the logarithm for the merit function, we get

$$\begin{aligned} & \underset{\boldsymbol{\beta}_{L\setminus N}}{\text{maximize}} \quad -\frac{1}{2} \cdot (\boldsymbol{\beta}_{L\setminus N} - \boldsymbol{\mu}_{\boldsymbol{\beta}_{L\setminus N}})^T \cdot \boldsymbol{\Sigma}_{\boldsymbol{\beta}_{L\setminus N}}^{-1} \cdot (\boldsymbol{\beta}_{L\setminus N} - \boldsymbol{\mu}_{\boldsymbol{\beta}_{L\setminus N}}) + \\ & \quad \sum_{k=1}^K \log \left\{ \mathbf{b}_{RAND,k} \cdot \mathbf{P}^{-T} \cdot \begin{bmatrix} \boldsymbol{\beta}_{L\setminus N} \\ 1 \end{bmatrix} \right\} \\ & \text{subject to} \quad \mathbf{B}_{GRID} \cdot \mathbf{P}^{-T} \cdot \begin{bmatrix} \boldsymbol{\beta}_{L\setminus N} \\ 1 \end{bmatrix} \geq 0 \end{aligned} \quad (38)$$

Studying (38) reveals two important observations. First, the merit function is the summation of a concave quadratic function and a number of concave logarithmic functions. Hence, the merit function is concave. Second, the constraint function is simply linear, implying that the constraint set is convex. For these reasons, the optimization in (38) is a convex programming problem and can be solved both efficiently (i.e., with low computational cost) and robustly (i.e., with guaranteed global optimum) by several numerical algorithms such as the interior point method [18]. Once the coefficients $\{\beta_{L,n}; n = 1, 2, \dots, N-1\}$ are found from (38), the coefficients $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ in (20) and, consequently, the late-stage performance distribution $pdf_L(g)$ in (2) can be easily determined.

3.4 Summary

Algorithm 1 summarizes the major steps of the proposed BMF method. Note that Algorithm 1 requires knowing the value of λ to define the prior distribution in (8)-(11). In practice, the appropriate value of λ can be determined by applying a cross validation method. These implementation details will be further discussed in Section 4.

Algorithm 1: Bayesian Model Fusion (BMF)

1. Start from a set of basis functions $\{b_n(g); n = 1, 2, \dots, N\}$, the early-stage coefficients $\{\alpha_{E,n}; n = 1, 2, \dots, N\}$, the late-stage simulation samples $\{g_{RAND,k}; k = 1, 2, \dots, K\}$, and a given value of λ .
2. Calculate the mean vector $\boldsymbol{\mu}_{\boldsymbol{\alpha}_L}$ and the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}_L}$ by using (10)-(11).
3. Construct the linear transformation in (14) based on the criteria specified in (20)-(21).
4. Calculate the mean vector $\boldsymbol{\mu}_{\boldsymbol{\beta}_L}$ and the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_L}$ by using (18)-(19).
5. Calculate the mean vector $\boldsymbol{\mu}_{\boldsymbol{\beta}_{L\setminus N}}$ and the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_{L\setminus N}}$ by using (25)-(27).
6. Construct the matrix \mathbf{B}_{GRID} in (31) and the vectors $\{\mathbf{b}_{RAND,k}; k = 1, 2, \dots, K\}$ in (36).
7. Solve the convex optimization problem in (38) to determine the solution $\boldsymbol{\beta}_{L\setminus N}$.
8. Calculate the coefficients $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ by using (20).
9. Determine the late-stage performance distribution functions $pdf_L(g)$ in (2) and $cdf_L(g)$ in (5).

4. IMPLEMENTATION DETAILS

The proposed BMF framework is made practically efficient by carefully tuning the Bayesian methodology to address a number of implementation issues. In this sub-section, we describe these implementation details, including (i) performance shift, and (ii) cross validation.

4.1 Performance Shift

As described in Section 3, one important assumption of BMF is that the early-stage and late-stage performance distributions, $pdf_E(g)$ and $pdf_L(g)$, are similar. In many practical applications, this assumption may not directly hold. For example, comparing the schematic-level and post-layout simulation results of an AMS circuit, we often notice that the nominal performance values are not matched and, hence, the corresponding performance distributions are significantly shifted due to device and interconnect parasitics. In this case, the early-stage and late-stage performance distributions are substantially different. Directly applying BMF to estimate the late-stage distribution $pdf_L(g)$ is unlikely to be accurate.

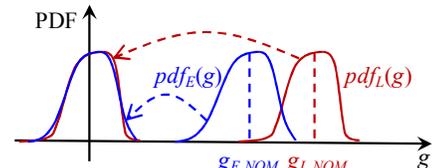


Figure 2. The early-stage and late-stage performance distributions $pdf_E(g)$ and $pdf_L(g)$ are shifted by the nominal performance values $g_{E,NOM}$ and $g_{L,NOM}$ respectively, before the proposed BMF method is applied.

To address this issue of performance shift, we propose to first

calculate the nominal performance values $g_{E,NOM}$ and $g_{L,NOM}$ by running one early-stage simulation and one late-stage simulation respectively. In practice, these nominal simulations are often performed by AMS designers, before they start to run statistical analysis to estimate the performance distributions. Once $g_{E,NOM}$ and $g_{L,NOM}$ are known, we shift the early-stage and late-stage performance distributions $pdf_E(g)$ and $pdf_L(g)$ by $g_{E,NOM}$ and $g_{L,NOM}$ respectively. As such, the shifted performance distributions are similar and the proposed BMF method becomes applicable. Figure 2 shows a simple example of the aforementioned performance shift for illustration purpose.

4.2 Cross Validation

To make the proposed BMF method of practical utility, we must appropriately determine the parameter λ that is used to define the prior distribution $pdf(\mathbf{a}_L)$ in (8)-(11). Note that the value of λ controls the *confidence* of our prior knowledge. If λ is small, the variance of the prior distribution $pdf(\mathbf{a}_L)$ is small, implying that the early-stage coefficients $\{\alpha_{E,n}; n = 1, 2, \dots, N\}$ closely approximate the late-stage coefficients $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$. Otherwise, if λ is large, the variance of the prior distribution $pdf(\mathbf{a}_L)$ is large and it is unlikely for the early-stage and late-stage coefficients to be similar. In practice, it is non-trivial to quantitatively determine the confidence (i.e., the value of λ) in advance. Instead, a smart algorithm must be used to automatically find the optimal λ based on a small number of late-stage simulation samples.

Towards this goal, we propose to repeatedly run the BMF flow with different values of λ to estimate the late-stage performance distribution and monitor the estimation accuracy. Note that the accuracy is expected to vary as λ varies. The optimal λ is determined by identifying the maximum accuracy being achieved and the corresponding λ value.

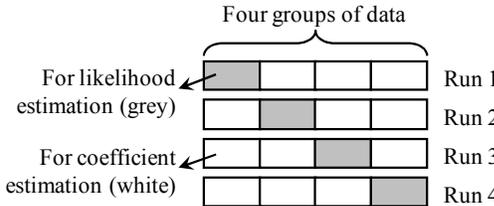


Figure 3. A simple four-fold cross validation example is shown for illustration purpose.

To assess the estimation accuracy for a given λ value, we adopt the idea of cross validation from the statistics community [20]. Given a set of late-stage simulation samples, a Q -fold cross validation partitions the entire data set into Q groups and repeatedly runs the BMF algorithm (i.e., Algorithm 1) for Q times. At each run, $Q - 1$ groups of simulation samples, referred to as the training set, are used to solve the late-stage coefficients $\{\alpha_{L,n}; n = 1, 2, \dots, N\}$ and estimate the late-stage performance distribution $pdf_L(g)$. The remaining group of simulation samples, referred to as the testing set, is used to assess the estimation accuracy based on the likelihood function in (34). If the estimated $pdf_L(g)$ is accurate, the likelihood function should be large. Namely, the probability of observing the simulation samples in the testing set should be large. Different groups should be selected as the testing set for different runs. As such, each run results in a likelihood value that is measured from a unique group of simulation samples. In addition, since the training set and the

testing set are separate in each run, over-fitting can be easily detected. Figure 3 shows a simple example for four-fold cross validation. More details about cross validation can be found in [20].

5. NUMERICAL EXAMPLES

In this section, several circuit examples designed in a commercial 32 nm CMOS process are used to demonstrate the efficiency of the proposed BMF method. For testing and comparison purposes, two different techniques are implemented to estimate the late-stage performance distributions: (i) the traditional kernel estimation method based on Gaussian kernel with optimal bandwidth [20], and (ii) the proposed BMF algorithm based on DCT (discrete cosine transform) basis functions [19]. In our experiments, CVX [21] is used to solve the convex programming problem in (38). All numerical experiments are run on a 2.9 GHz Linux server with 4 GB memory.

5.1 Ring Oscillator

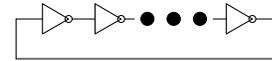


Figure 4. A simplified circuit schematic is shown for a ring oscillator (RO) designed in a commercial 32 nm CMOS process.

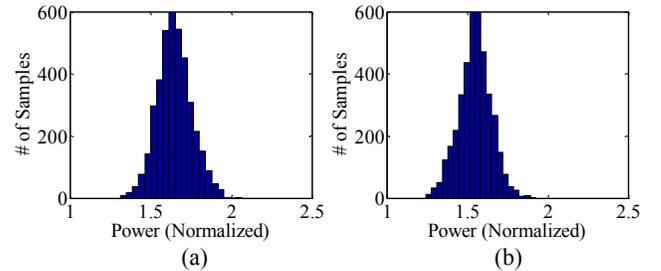


Figure 5. Histograms of RO power are shown for (a) schematic-level and (b) post-layout simulation data.

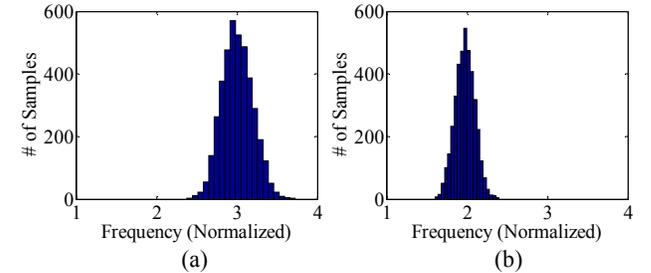


Figure 6. Histograms of RO frequency are shown for (a) schematic-level and (b) post-layout simulation data.

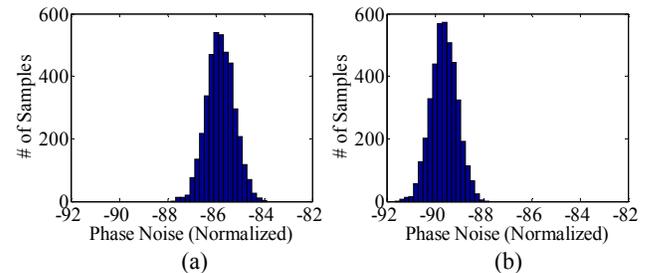


Figure 7. Histograms of RO phase noise are shown for (a) schematic-level and (b) post-layout simulation data.

Shown in Figure 4 is the simplified circuit schematic of a ring oscillator (RO) designed in a commercial 32 nm CMOS process. We consider three performance metrics: power, frequency and phase noise. In this example, 4000 random samples are generated by both schematic-level and post-layout simulations. The runtime to generate a single schematic-level simulation sample is 5.32 seconds, and it is 45.43 seconds for post-layout simulation. Figure 5-Figure 7 show the histograms for different performance metrics estimated by all simulation samples. Note that the mean values of the schematic-level and post-layout distributions are often substantially different.

Figure 8(a)-Figure 10(a) plot the cumulative distribution functions estimated from 40 post-layout simulation samples by different methods. Here, the exact distributions are calculated from all 4000 simulation samples and they are used to compare the estimation accuracy. Given the same number of post-layout simulation samples, BMF achieves superior accuracy over the traditional kernel estimation in this example.

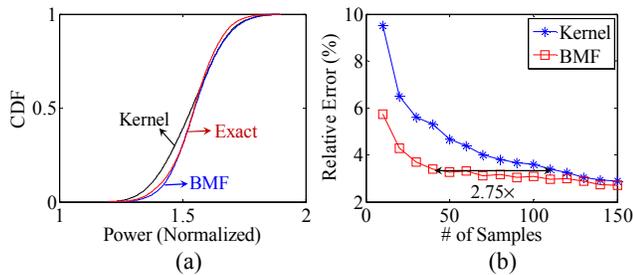


Figure 8. (a) Cumulative distribution functions of RO power are estimated by 40 post-layout simulation samples. (b) Estimation error of RO power distribution is shown for different methods.

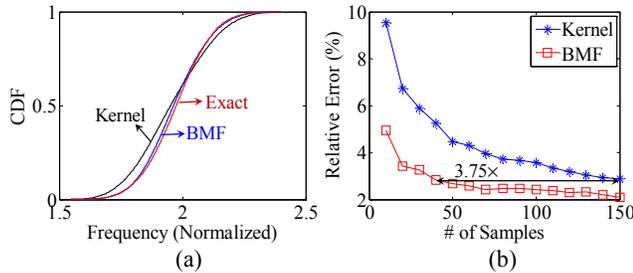


Figure 9. (a) Cumulative distribution functions of RO frequency are estimated by 40 post-layout simulation samples. (b) Estimation error of RO frequency distribution is shown for different methods.

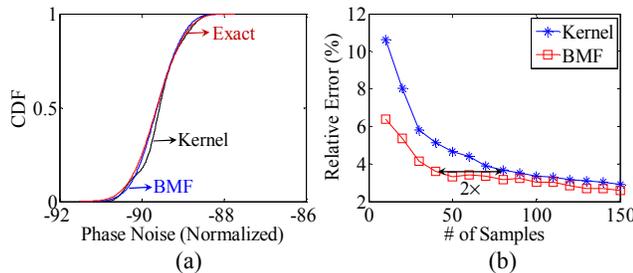


Figure 10. (a) Cumulative distribution functions of RO phase noise are estimated by 40 post-layout simulation samples. (b) Estimation error of RO phase noise distribution is shown for different methods.

Figure 8(b)-Figure 10(b) further quantitatively show the estimation error for different methods. Here, the estimation error is calculated for the cumulative distribution functions

$$\sqrt{\int_{g_{MIN}}^{g_{MAX}} [cdf_L(g) - \tilde{cdf}_L(g)]^2 \cdot dg} / \sqrt{\int_{g_{MIN}}^{g_{MAX}} [cdf_L(g)]^2 \cdot dg} \quad (39)$$

where $cdf_L(g)$ and $\tilde{cdf}_L(g)$ denote the exact and estimated cumulative distribution functions respectively. Note that the traditional kernel estimation requires up to 3.75 \times more simulation samples than BMF to achieve the same accuracy. In other words, BMF achieves up to 3.75 \times runtime speedup over kernel estimation in this example. For instance, BMF reduces the total post-layout simulation time from 1.89 hours (150 samples) to 30.29 minutes (40 samples) when estimating the RO frequency distribution, as shown in Figure 9(b). Once the simulation samples are available, running the BMF algorithm (i.e., Algorithm 1) only takes a few seconds and, hence, its computational time is negligible.

5.2 SRAM Read Path

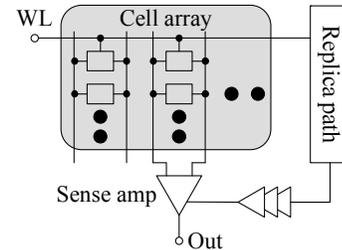


Figure 11. A simplified circuit schematic is shown for an SRAM read path designed in a commercial 32 nm CMOS process.

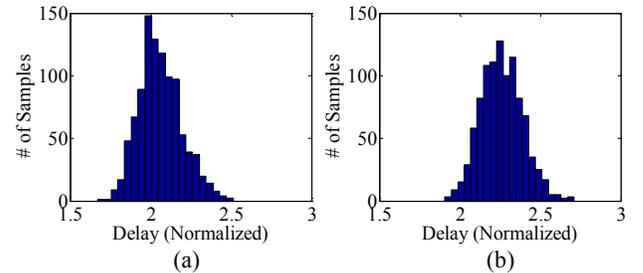


Figure 12. Histograms of SRAM read path delay are shown for (a) schematic-level and (b) post-layout simulation data.

Figure 11 shows the simplified circuit schematic of an SRAM read path designed in a commercial 32 nm CMOS process. In this example, we are interested in the read path delay from the word line (WL) to the sense amplifier output (Out). For testing and comparison purpose, 1000 random samples are generated by both schematic-level and post-layout simulations. The runtime to generate a single schematic-level simulation sample is 26.44 seconds, and it is 104.58 seconds for post-layout simulation. Figure 12 shows the histograms for both schematic-level and post-layout performance distributions estimated from all 1000 simulation samples.

Figure 13(a) plots the cumulative distribution functions estimated from 40 post-layout simulation samples by different methods. Similar to the previous RO example, BMF offers superior accuracy over the traditional kernel estimation. Figure 13(b) further shows the log-likelihood value (normalized)

estimated by cross validation for different values of λ . It can be seen that the likelihood function reaches its maximum at a particular λ (i.e., the optimal λ value).

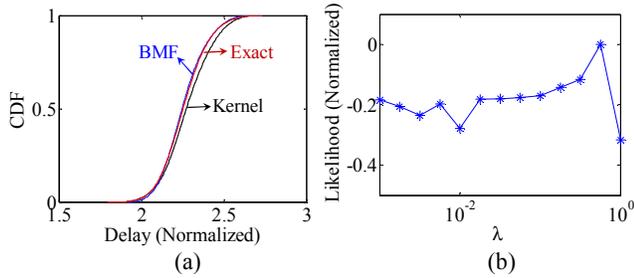


Figure 13. (a) Cumulative distribution functions of SRAM read path delay are estimated by 40 post-layout simulation samples. (b) Log-likelihood value (normalized) is shown as a function of λ .

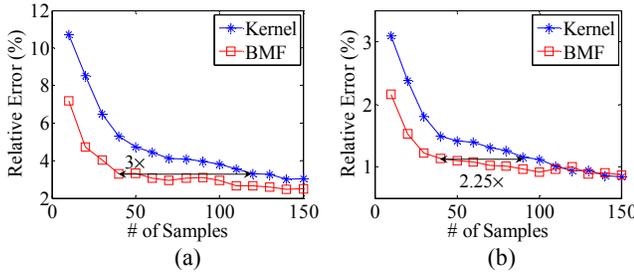


Figure 14. (a) Estimation error of SRAM read path delay distribution is shown for different methods. (b) Estimation error of worst-case SRAM read path delay is shown for different methods.

Figure 14(a) shows the estimation error for the cumulative distribution functions and Figure 14(b) shows the relative error for the estimated worst-case delay. Here, the worst-case delay is defined as the 95% point of the corresponding cumulative distribution function. Studying both plots reveals that BMF achieves up to 3 \times runtime speedup over kernel estimation in this example. It reduces the total post-layout simulation time from 3.49 hours (120 samples) to 1.16 hours (40 samples).

6. CONCLUSIONS

In this paper, a novel BMF technique is proposed to efficiently estimate the performance distributions of analog and mixed-signal circuits. BMF borrows the simulation data from an early stage to predict the performance distributions at a late stage. In other words, it aims to reduce the number of required late-stage simulation runs by fusing the early-stage and late-stage performance distributions through Bayesian inference. Several circuit examples designed in a commercial 32 nm CMOS process demonstrate that the proposed BMF technique achieves up to 3.75 \times runtime speedup over the traditional kernel estimation method. Finally, it is important to mention that even though BMF is proposed for pre-silicon parametric yield estimation in this paper, it can be further extended to estimate the performance distributions based on post-silicon measurement data. The possible applications of BMF for AMS post-silicon validation will be studied in our future research.

7. ACKNOWLEDGEMENTS

The authors acknowledge the support of the C2S2 Focus Center, one of six research centers funded under the Focus Center

Research Program (FCRP), a Semiconductor Research Corporation entity. This work is also supported in part by the National Science Foundation.

8. REFERENCES

- [1] S. Nassif, "Modeling and analysis of manufacturing variations," *IEEE CICC*, pp. 223-228, 2001.
- [2] X. Li, J. Le and L. Pileggi, *Statistical Performance Modeling and Optimization*, Now Publishers, 2007.
- [3] Semiconductor Industry Associate, *International Technology Roadmap for Semiconductors*, 2011.
- [4] M. Styblinski and J. Zhang, "Circuit performance variability reduction: principles, problems, and practical solutions," *IEEE ICCAD*, pp. 170-173, 1991.
- [5] Z. Wang and S. Director, "An efficient yield optimization method using a two step linear approximation of circuit performance," *IEEE EDAC*, pp. 567-571, 1994.
- [6] A. Dharchoudhury and S. Kang, "Worse-case analysis and optimization of VLSI circuit performance," *IEEE Trans. on CAD*, vol. 14, no. 4, pp. 481-492, Apr. 1995.
- [7] E. Felt, S. Zanella, C. Guardiani and A. Sangiovanni-Vincentelli, "Hierarchical statistical characterization of mixed-signal circuits using behavioral modeling," *IEEE ICCAD*, pp. 374-380, 1996.
- [8] G. Debyser and G. Gielen, "Efficient analog circuit synthesis with simultaneous yield and robustness optimization," *IEEE ICCAD*, pp. 308-311, 1998.
- [9] 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.
- [10] X. Li, P. Gopalakrishnan, Y. Xu and L. Pileggi, "Robust analog/RF circuit design with projection-based performance modeling," *IEEE Trans. on CAD*, vol. 26, no. 1, pp. 2-15, Jan. 2007.
- [11] F. Gong, H. Yu and L. He, "Stochastic analog circuit behavior modeling by point estimation method," *IEEE ISPD*, pp. 175-182, 2011.
- [12] X. Li, J. Le, P. Gopalakrishnan and L. Pileggi, "Asymptotic probability extraction for nonnormal performance distributions," *IEEE Trans. on CAD*, vol. 26, no. 1, pp. 16-37, Jan. 2007.
- [13] X. Li, Y. Zhang and L. Pileggi, "Quadratic statistical MAX approximation for parametric yield estimation of analog/RF integrated circuits," *IEEE Trans. on CAD*, vol. 27, no. 5, pp. 831-843, May. 2008.
- [14] P. Beckmann, *Orthogonal Polynomials for Engineers and Physicists*, Golem Press, 1973.
- [15] A. Oppenheim, R. Schafer and J. Buck, *Discrete-Time Signal Processing*, 1999.
- [16] A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 2001.
- [17] R. Myers and D. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley-Interscience, 2002.
- [18] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [19] R. Gonzalez and R. Woods, *Digital Image Processing*, Prentice Hall, 2007.
- [20] C. Bishop, *Pattern Recognition and Machine Learning*, Prentice Hall, 2007.
- [21] M. Grant and S. Boyd, *Matlab Software for Disciplined Convex Programming* (<http://stanford.edu/~boyd/cvx>), Jun. 2009.