Accurate Multi-segment Probability Density Estimation Through Moment Matching

Rahul Krishnan, Wei Wu, *Student Member, IEEE*, Srinivas Bodapati, *Member, IEEE*, and Lei He, *Senior Member, IEEE*

***Abstract*—The impact of process variations continues to grow as transistor feature size shrinks. Such variations in transistor parameters lead to variations and unpredictability in circuit output and may ultimately cause them to violate specifications leading to circuit failure. In fact, timely failures in critical circuits may lead to catastrophic failures in the entire chip. As such, statistical modeling of circuit behavior is becoming increasingly important. However, existing statistical circuit simulation approaches fail to accurately and efficiently analyze the high sigma behavior of probabilistic circuit output. To this end, we propose PDM (Piecewise Distribution Model) - a piecewise distribution modeling approach via moment matching using maximum entropy to model the high sigma behavior of analog/mixed-signal (AMS) circuit probability distributions. PDM is independent of the number of input dimensions and matches region specific probabilistic moments which allows for significantly greater accuracy compared to other moment matching approaches. PDM also utilizes Spearman's rank correlation coefficient to select the optimal approximation for the tail of the distribution. Experiments on a known mathematical distribution and various circuits obtain accurate results up to 4.8 sigma with 2-3 orders of speedup relative to Monte Carlo. PDM also demonstrates better accuracy while compared against other state-of-the-art statistical modeling approaches, such as maximum entropy, importance sampling, and subset simulation.**

***Index Terms*—Moment matching, High dimensional, Maximum Entropy, Probability density function, Circuit modeling**

I. Introduction

As transistor feature size continues to shrink, the impact of process variations on circuit behavior, such as delay or gain, grows and cannot be neglected [1], [2], [3], [4], [5]. Under these variations, circuit behavior is no longer a deterministic value and must be characterized by a random variable rather than a nominal value. These variations can cause significant circuit performance degradation that may not meet the design spec and fail. As such, circuit reliability has become an area of growing concern. In particular, for circuits that are repeated millions of times, a small failure probability may lead to catastrophic results in the entire chip. Consequently, such "rare event" failures must be accurately and efficiently modeled to maximize the effective yield of a circuit.

As industry moves towards more energy efficient chips, minimizing power consumption becomes increasingly important. In such designs, low supply voltages (VDD) are often used to reduce power. However, while VDD is explicitly reduced the overdrive voltage ($V_{gs} - V_{th}$) is implicitly reduced [6]. In the presence of V_{th} variations from the manufacturing process, transistors may enter the subthreshold operation region causing a strongly non-linear circuit behavior. This non-linear behavior translates to circuit behavior distributions becoming strongly non-Gaussian (see Fig. 14). Consequently, when modeling this behavior for yield analysis, it is necessary to consider the inherent non-linearity that arises due to the aforementioned reasons.

Although there are many methods that attempt to model overall circuit behavior [1], [3], [7], very few of them efficiently model the high sigma behavior of strongly non-Gaussian distributions. One brute force method is Monte Carlo (MC), which is considered to be the gold standard approach; it involves repeated sampling and simulation to extract an approximate distribution of circuit behavior [8]. Although Monte Carlo is highly accurate, it is infeasible for yield analysis with small failure probability because it requires millions of samples/simulations for an accurate measurement, making it runtime prohibitive despite some parallelization efforts [9], [10], [11]. Moreover, if any design changes are introduced in the circuit we must repeat these simulations another million or more times.

In order to improve the efficiency of yield analysis, fast MC approaches such as Importance Sampling (IS) [12], [13], [14], [15], [16], [17], [18], and classification based approaches [19], [20] were proposed to obtain high accuracy with a minimal number of samples. However, Importance Sampling methods can rarely handle circuits with a large number of variables due to the "curse of dimensionality" which causes the reweighing process to become degenerate and unbounded [21], [22]. Classification based approaches, such as statistical Blockade [19], [23], attempt to build a linear classifier to screen out/block samples that are unlikely to cause failure, and evaluate these "likely-to-fail" samples to calculate a failure probability. However, the classifier used in Statistical Blockade [19], [23] does not account for the non-linearity between process variables and circuit outputs, or the multiplicity of input failure regions, leading to large errors. The nonlinear

R. Krishnan and W. Wu are with the Department of Electrical Engineering, University of California, Los Angeles (email: [r.krishnan390, weiwu2011]@ucla.edu)

L. He is with China State Key Laboratory on Application Specific IC and Systems at Fudan University and the Department of Electrical Engineering, University of California, Los Angeles (email: [lhe]@ee.ucla.edu)

S. Bodapati is with Intel Corp., Santa Clara, CA (email: [srinivas.bodapati]@intel.com)

Manuscript received: June 20, 2015; revised: September 19, 2015 and December 12, 2015; accepted: Apri 9, 2016.

This work was partially supported by Intel and China State Key Laboratory on Application Specific IC and Systems at Fudan University. Lei He's portion of work was performed during his visit at Fudan.

classifier separates the "likely-to-fail" samples with better accuracy [20], but has difficulty in accurately defining the "likely-to-fail" samples. On the other hand, defining only a small portion of samples as "likely-to-fail" leads to skewed classes that require separation - while adding more samples to the "likely-to-fail" side balances the classes, it could make classification based approaches inefficient.

Among others, the Scaled Sigma Sampling (SSS) [24] and subset simulation (SUS) [25] approach the rare failure probability via different avenues. SSS draws samples by scaling up the standard deviation (sigma) of the original distribution, while using the same mean. Failure probabilities are calculated at different scaling factors to extrapolate the failure probability under the original distribution, i.e. scaling factor equal to 1 [24]. However, SSS is susceptible to accuracy loss due to the extrapolation requirement. Alternatively, SUS approaches the rare failure probability as the production of several, large conditional probabilities estimated in multiple phases [25]. Samples in each phase are generated with the aid of the Markov Chain Monte Carlo (MCMC) method.

Unfortunately, the majority of existing approaches do not efficiently handle a significantly high dimensional problem. While [12], [13], [14], [15], [16] are only verified on SRAM cells with 6 variation parameters, there are still some approaches that do handle the high dimensional problem, such as [17], [24], [25]. Furthermore, most existing approaches do not estimate the overall PDF of circuit behavior, requiring repetitive sampling to estimate different critical points causing significant runtime overhead.

To combat the dimensionality issue of the above methods, we introduced a moment matching technique based on Maximum Entropy [26] (referred to as MAXENT), which is elaborated upon in Section III. The method is novel because it uses circuit output behavior (e.g. delay) as its only input and therefore performs moment matching solely in the output domain. Consequently, the method is constant in dimensionality and thus does not fall to the dimensionality issues in Importance Sampling and classifier methods outlined above. We observed that MAXENT is very accurate in the bulk of the distribution, but is often inaccurate in the tail region where rare events are modeled. This limitation is because MAXENT uses only one set of moments that are accurate in the low sigma region but inaccurate in the tail. Obtaining moments that are accurate in the tail of the distribution (also known as the high sigma region) requires both a large number of samples to obtain accurate moments and knowledge of which exact moments reflect behavior in the tail of the distribution, which is often unknown [27]. Consequently, the distribution that MAXENT uses is formulated on a global optimization framework that attempts to minimize overall error, making it difficult to capture the high sigma behavior in non-Gaussian distributions.

To address both the issue of high-dimensionality *and* non-Gaussian distributions while maintaining high accuracy and efficiency, we propose a piecewise distribution model (PDM) that uses moment matching via maximum entropy to build multiple separate, region-based distributions of circuit behavior. Without loss of generality, we consider a distribution as two segments in the rest of this paper. The first distribution, Segment1, matches moments that are accurate only in the body/bulk of the distribution. The second distribution, Segment2, matches moments that are accurate only in the high sigma/tail region of the distribution and models the tail of circuit behavior. Both distributions are constructed using the maximum entropy moment matching technique but differ by using two *different sets* of moments. The moments in Segment1 are obtained by using circuit behavior sample moments calculated directly from the original input (process variation) distributions. The moments in Segment2 are obtained using sample moments calculated from input distributions that are *shifted* towards regions that are more likely to fail. The details of moment calculation are elaborated upon in Section IV.

The optimal Segment1 distribution is selected using Spearman's rank correlation coefficient to analyze the monotonic behavior of the CDF. The Segment2 distribution is assumed to be an exponential distribution. Because this distribution is constructed from shifted moments, its probability must be reweighed and is done so using conditional probability and a scaling factor that corrects for continuity between the Segment2 distribution and the true model of the tail distribution.

PDM has a constant complexity in terms of input dimensions as it works solely in the output (circuit behavior) domain. Experiments on both a mathematically known distribution and circuits demonstrate the method is accurate up to 4.8 sigma for non-Gaussian distributions with more than 2 orders of speedup relative to Monte Carlo, which is typically sufficient for analog circuits that are reused, such as differential amplifiers, bias circuit, or even PLLs, level shifters, etc.

The performance of PDM is compared against the maximum entropy moment matching technique [26], high dimensional importance sampling [17], SUS [25], and Monte Carlo. The statistical modeling approaches are compared on both low-dimensional and high-dimensional problems, in addition to ideal (mathematical) distributions and realistic circuits. Runtime is evaluated by the number of samples required by each algorithm, and accuracy is compared against Monte Carlo.

The rest of the paper is organized as follows. Section II presents background on general statistical modeling. Section III introduces a detailed derivation of the maximum entropy moment matching technique, along with results on its application to circuit modeling [26]. Section IV presents the proposed piecewise distribution model and highlights the difference between it and the general maximum entropy moment matching technique. Section V evaluates the performance of PDM on the mathematically known distribution, one digital circuit, and one analog circuit, where all distributions are non-Gaussian. Section VI concludes this paper and presents some topics for future work.

II. Background

A. General Statistical Modeling

Fig. 1 shows the flow of performing statistical simulation on a circuit. Instead of simulating a circuit with deterministic parameters a single time, we model the circuit parameters, such as effective channel length L_{eff} or oxide thickness t_{ox},

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2562923, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

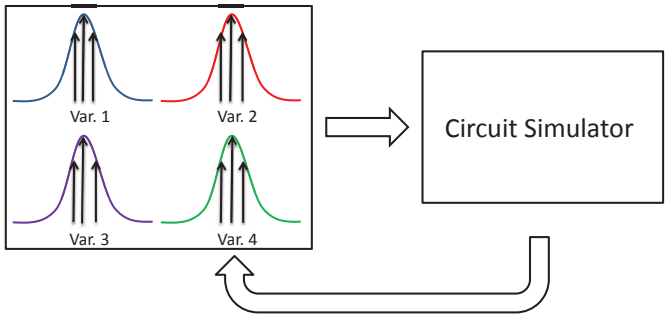


Fig. 1. Flowchart for Circuit Simulation

as random variables. These variables form the "parameter domain" as the input of the statistical simulation. Without losing generality, we model these variations as independent Gaussian distributions.

Given N input random variables, we draw a single sample x_1 which can be represented as the vector $x_1 = [x_{1,1}, x_{1,2}, ..., x_{1,N}]$. By feeding sample x_1 to a SPICE-accurate circuit simulator, we can obtain a circuit response y_1. The circuit simulator acts as a non-linear map from input samples to circuit response. These circuit responses, such as 50% delay or voltage at a node, are considered part of the "output domain".

Repeating this sampling and simulation process is straightforward Monte Carlo and is used to obtain an estimate of the distribution/probability density function of circuit response [8]. Depending on the required accuracy of the response, the required number of samples will change. In some cases, a few hundred samples are required to estimate the probability of a circuit response around the nominal value of the PDF. On the other hand, it requires millions or even more samples to model the tail of the distribution, corresponding to the rare failure events, which is important for highly replicated circuit cells or critical circuit components.

One method of quantifying the required number of samples for a target probability is simply taking the inverse. For example, consider a designer that is interested in the circuit response that will result in a failure rate of 16%. This means that we are interested in a circuit response that has approximately 16% probability in the tail of the PDF. This failure rate corresponds to approximately 1 failure every 6.25 samples, so a starting point would be drawing 7 samples, simulating each and selecting the largest value. However, the preceding case assumes that we will determinately see 1 failed sample every 6.25 which may not be true due to the large variations and unpredictability in the circuit. Consequently, in order to have a more confident estimate, we may require that we draw enough samples such that we have 5 failures, i.e. we would draw 32 samples. By using basic probability, we do not make any assumptions about the shape of the circuit PDF allowing for an unbiased estimate. Furthermore, to simplify the relationship between estimated probability (failure rate) and required number of samples, we utilize the Z score of a standard normal distribution which is typically referred to as the "sigma" value [27]. For example, instead of asking for the circuit response that gives a 16% failure rate, we would ask for the "1 sigma point" of the PDF. Note that the aforementioned probability and Z score methods work for both ends of the tail of a PDF.

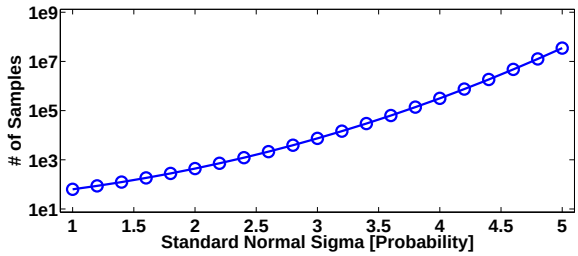


Fig. 2. Required samples for various sigma (probability) points

In typical yield analysis, we require a failure rate of at least 0.003167% or approximately 4 sigma. Obtaining this probability would require one failure every $31,575$ samples. In many cases, this is the "largest" probability that is required and we typically need up to the 4.8 or 5 sigma point in the PDF, which corresponds to roughly $3.4E6$ samples for one failure. Furthermore, simulating this many samples is overly time consuming, making straightforward Monte Carlo runtime prohibitive. Fig. 2 shows the required number of samples (log scale) for a given sigma point (linear). We see that even in log scale, the required number of samples is non-linear. Although more efficient sampling approaches, e.g. Quasi Monte Carlo, Latin Hypercube Sampling, etc. [28], might be used, they are still insufficient to analyze the rare failure event.

Alternative methods such as Quasi Monte Carlo may be utilized, however they still require a large number of samples and it can be shown that as the dimensionality of the sampling space increases (in this case, the input domain), the convergence rate of QMC and MC are similar [28]. Consequently, it is necessary to develop efficient algorithms that minimize the number of samples to accurately estimate a very small failure probability.

III. Maximum Entropy Method for Statistical Circuit Performance Modeling

Entropy is a measure of uncertainty. When choosing a distribution, one should choose a distribution that maximizes the entropy [29]. By doing this, the distribution is uniquely determined to be maximally unbiased with regard to missing information, while still agreeing with what is known [29]. Consequently, the distribution with the maximum entropy will create a model based solely on the true information that is provided and will be less susceptible to assumptions from missing information. The entropy W of a distribution $p(x)$ is defined in (1). To select the distribution with the least missing information, we maximize the entropy function with respect to a set of probabilistic moment constraints (2), as a probability distribution can be completely defined by its set of moments [30]. When applying the maximum entropy method to circuit simulation algorithms, we consider probabilistic sample moments of circuit response with moment order $i = 0, 1, ..., k$.

$$W = \int -p(x)\log p(x)dx \tag{1}$$

$$\int x^i p(x)dx = \mu_i, \qquad i = 0, 1, ..., k. \tag{2}$$

To maximize (1) we first introduce Lagrange multipliers, resulting in the Lagrangian function

$$L = -\int (p(x)\log p(x))dx + \sum_{i=0}^{k} \lambda_i (\int x^i p(x)dx - \mu_i) \tag{3}$$

Next, we take partial derivatives of L with respect to $p(x)$ and λ to find the points where it reaches a maximum, as shown in (5) and (4).

$$\frac{\delta L}{\delta \lambda_i} = 0 \tag{4}$$

$$\frac{\delta L}{\delta p(x)} = 0 \tag{5}$$

Taking the derivative with respect to λ results in the original moment constraints from (2) and is redundant information. The derivative with respect to $p(x)$ yields (6)

$$\frac{\delta L}{\delta p(x)} = \int (\log p(x)dx) + 1 - \{\sum_{i=0}^{k} \lambda_i (\int (x^i dx))\} = 0 \tag{6}$$

We can further simplify this by absorbing the constant 1 into the λ_0 term and combining the finite sum with the integrand resulting in (7)

$$\int (\log p(x)dx) - \int (\sum_{i=0}^{k} \lambda_i x^i dx) = 0 \tag{7}$$

Note that the limits on both integrals are identical and are typically from ∞ to $-\infty$ for standard probability distributions because the distribution is assumed to be 0 outside of the support of random variable x. In the case of circuit simulation algorithms, this is also true, i.e. the circuit has maximum and minimum operating values and is zero outside these points. Consequently, because the above equation must hold in the general case of arbitrary limits, the integrand must be 0 and we can rearrange terms to solve for the unknown variable $p(x)$ as shown in (8).

$$p(x) = \exp\left(-\sum_{i=0}^{k} \lambda_i x^i\right) \tag{8}$$

However, the solution in (8) does not exist for values of $k \geq 2$ [31]. Consequently, [32] propose that we transform the constrained problem into an unconstrained problem by utilizing its dual. Utilizing duality allows us to recast the original problem of maximizing (3) into its dual form that we can minimize. This dual function can be obtained by plugging the results of (8) into the Lagrange function (3) resulting in its dual, which is represented by the two functions (9) and (10). We also note that the term μ_0 is simply a normalizing factor representing the area and is thus folded into equations (9) and (10). For further details in the derivation, we refer the reader towards [33].

$$\Gamma = \ln Z + \sum_{i=1}^{k} \lambda_i \mu_i \tag{9}$$

$$Z = \exp(\lambda_0) = \int \exp\left(-\sum_{i=1}^{k} \lambda_i x^i\right) dx \tag{10}$$

Now this dual problem can be solved for any value of k. One approach is using an iterative method such as traditional Newton's method as shown in [31], [34], [26]. Here, Newton's method is used to solve for the Lagrangian multipliers $\lambda = [\lambda_0, \lambda_1, ..., \lambda_k]'$ for a corresponding set of moments $i = 0, 1, ..., k$. The standard Newton update equation for iteration m is shown in (11)

$$\lambda_{(m)} = \lambda_{(m)} - H^{-1}\frac{\delta \Gamma}{\delta \lambda} \tag{11}$$

Where the gradient (12) and Hessian (13) are defined as

$$\frac{\delta \Gamma}{\delta \lambda_i} = \mu_i - \frac{\int x^i \exp\left(-\sum_{i=1}^{k} \lambda_i \mu_i\right) dx}{\int \exp\left(-\sum_{i=1}^{k} \lambda_i \mu_i\right) dx} = \mu_i - \mu_i(\lambda) \tag{12}$$

$$H_{ij} = \frac{\delta^2 \Gamma}{\delta \lambda_i \delta \lambda_j} = \mu_{i+j}(\lambda) - \mu_i(\lambda)\mu_j(\lambda) \tag{13}$$

$$\mu_{i+j}(\lambda) = \frac{\int x^{i+j} \exp\left(-\sum_{i=1}^{k} \lambda_i \mu_i\right) dx}{\int \exp\left(-\sum_{i=1}^{k} \lambda_i \mu_i\right) dx} \tag{14}$$

Equation (13) indicates that the dual function Γ has a second derivative and that it is positive definite [33]. Consequently, the function (9) is everywhere convex which guarantees that if a stationary point exists it must be the *unique absolute minimum*. However, convexity does not ensure that a minimum does exist. Consequently, a necessary and sufficient condition that a unique absolute minimum exists at a finite value of λ is that the moment sequence $\{\mu_i, i = 0, 1, ..., k\}$ be completely monotonic [33]. We note that the derivation of such an existence condition is outside of the circuit simulation topic and we therefore refer to [33] for its derivation.

A. Application to Statistical Circuit Modeling

We begin by drawing a small number of samples from the input variables and feeding them to a circuit simulator to produce a set of outputs. These small number of outputs are realizations of the random variable x which are used to construct the moments μ_i that are to be matched in the optimization. Note that probabilistic moments are typically calculated as $\mu_i = \int x^i p(x)dx$. However, because we have no prior information about the shape or form of the distribution $p(x)$, we cannot use this method. Consequently, we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2562923, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

utilize sample moments $\mu_i = \sum_{j=1}^{N} x_j^i/N$ (N is the number of samples that we draw) to construct the moments for this generic case[27]. By using sample moments, we ensure that the requirement for monotonic moments is satisfied because the random variable x is assumed to be always positive (we can always transform the circuit response to be positive). Consequently, we are guaranteed that the estimated probability distribution $p(x)$ will be stable.

After obtaining the sample moments μ_i for a set $i = 0, 1, ...k$ we perform the maximum entropy moment matching method using traditional Newton's method. We initialize the Lagrange multipliers to 0, $\lambda = [0; 0; ...; 0]$, resulting in the initial guess of the distribution as a uniform distribution. This result is reasonable as the uniform distribution inherently has the maximum entropy of all distributions. Next, we let the algorithm continue until the successive changes in multipliers λ_i are within a user specified tolerance. As such, we obtain a probability distribution $p(x)_k$ where k denotes the number of moments that are used.

B. Preliminary Results using MAXENT

Examples of this work are implemented as the MAXENT algorithm and are shown in [26]. We implemented the proposed algorithm in MATLAB. The first circuit is a 6-T SRAM bit-cell with 54 variables, while the second circuit is an Operational Amplifier with 70 variables. HSPICE is used to simulate these 2 circuits for circuit performance. Also, MC [8] and PEM [1] are used for comparison. PEM is another circuit modeling algorithm that converts probabilistic moments of circuit performance into corresponding time moments of an LTI system then uses Asymptotic Waveform Evaluation (AWE) to match these time moments to the transfer function of the system. AWE uses the Pade approximation which generates poles (eigenvalues) that correspond to the dominant poles of the original system, and also poles that do not correspond to the poles of the original system but account for the effects of the remaining poles [35].

Fig. 3 shows a schematic of the 6T SRAM bit cell. The reading operation of the cell can either be a success or a failure based on the voltage difference ΔV between the nodes BL and $\overline{BL}$. If the voltage ΔV is large enough to be sensed, then the reading operation is considered to be successful. Due to the process variations in every transistor, the discharge behavior of $\overline{BL}$ will vary for different cells and conditions. Consequently, the designed discharge behavior will have significant variation, and if the behavior is drastically different, the voltage ΔV may not be sufficiently large causing a read failure. Fig. 4 shows a schematic of the Operational Amplifier that was used. We considered the bandwidth of the amplifier as the circuit performance to be modeled.

C. Stability

The stability of the two algorithms is clearly demonstrated in Fig. 5a, Fig. 5b, and 5c which show the performance distributions that model the value of ΔV in the 6T SRAM circuit. In all 3 figures, both MAXENT and PEM were

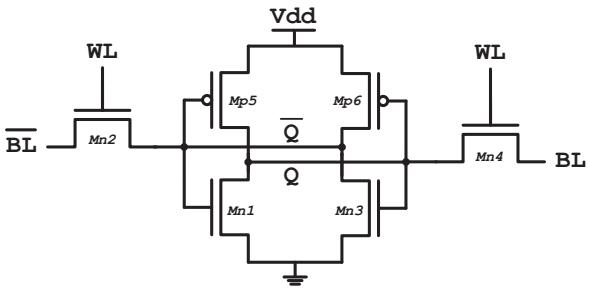


Fig. 3. 6T SRAM Circuit Layout

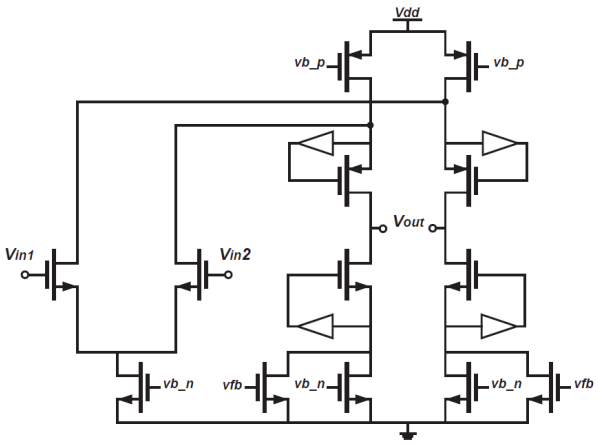
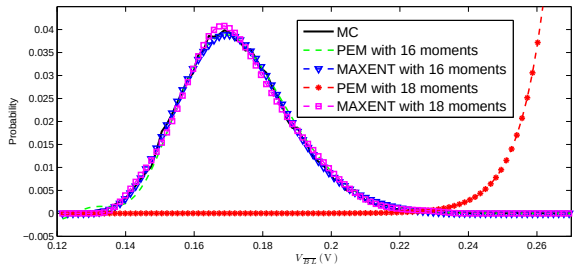


Fig. 4. Operational Amplifier Circuit Layout

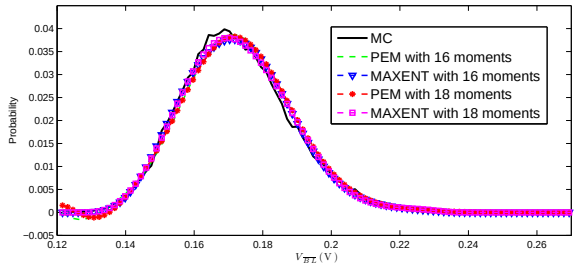
constructed using the same set of moments. We can see that PEM is clearly dependent on both the number of moments and number of samples used. In Fig. 5 PEM is unstable when using 18 moments constructed with 200 samples, but becomes stable in Fig. 5b when using 18 moments constructed with 250 samples. However, PEM becomes unstable again in Fig. 5c when using 18 moments constructed with 300 samples. Clearly, increasing the number of samples has produced a set of moments that allows PEM to be stable under scenarios where it was previously unstable. On the other hand, we can see that MAXENT is stable in all 3 figures regardless of the number of moments used or the number of samples used to construct them. Furthermore, the MAXENT distributions show very good overlap with the Monte Carlo distributions.

PEM's sensitivity arises because it uses the Pade approximation, which can produce small, positive poles, to estimate a transfer function model that is used to match moments. Since PEM uses a transfer function in its model for the distribution, these positive poles can produce the instability we see above. On the other hand, MAXENT is stable in all of the previous cases. MAXENT is guaranteed to be stable if it uses monotonic moments and its stability is not sensitive to the order of moments that are matched or the number of samples that are used to produce these moments. Moreover, because MAXENT estimates its distribution as a product of exponential functions, it will never have a negative probability. Consequently, we see that MAXENT is more robust when

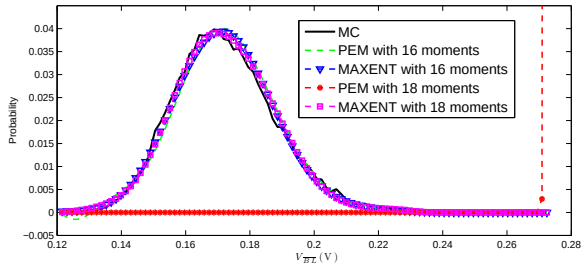
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2562923, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



(a) PEM lack of stability on SRAM circuit (200 samples)



(b) PEM stability on SRAM circuit (250 samples)



(c) PEM lack of stability on SRAM circuit (300 samples)

Fig. 5. Stability of PEM and MAXENT under different number of samples

compared to other moment matching methods such as PEM.

D. Accuracy

Fig. 6 shows the different distributions generated by MAXENT and PEM vs the ground truth distribution from MC for the Operational Amplifier circuit. We see that using 10 moments, MAXENT does a good job of estimating the overall shape of the distribution but still lacks some detail. Increasing the order of moments to 12 produces a distribution that overlaps extremely well with the ground truth distribution. On the other hand, PEM fails to give an accurate estimate of the distribution with both 10 and 12 moments. When moving from 10 to 12 moments with MAXENT, we saw a significant increase in accuracy. When moving from 10 to 12 moments with PEM, we see essentially no change in accuracy. Furthermore, we see that PEM produces an unreasonable, negative probability in its distribution.

To quantify the accuracy results, Tables I and II shows the relative error (calculated by (15)) for both MAXENT and PEM in the 6T SRAM and Operational Amplifier circuits. As we can see from both tables, MAXENT offers up to 110% less error for the OpAmp, and up to 27% less error for the SRAM circuit once we reach a steady-state value. We also note that although values of variance and kurtosis (moments 2 and 4) are accurately calculated, the distributions generated using only 2 and 4 moments are inaccurate. Distributions generated with 2 and 4 moments are identical to an Exponential and Gaussian distribution, respectively, due to the mathematical representation of the maximum entropy distribution [36]. This type of inaccuracy is present in all moment matching algorithms and these two orders were therefore excluded from the accuracy tables.

$$error = \int (f_1(x) - f_2(x))\, dx \tag{15}$$

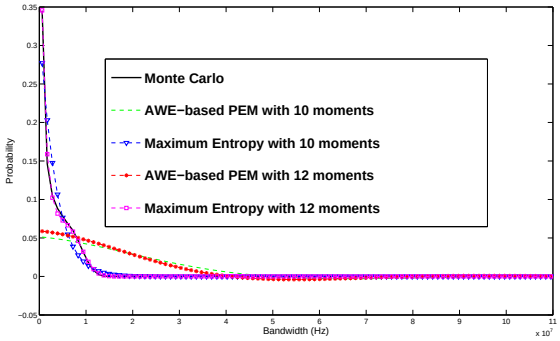


Fig. 6. Operational Amplifier Accuracy (800 samples)

TABLE I
ACCURACY COMPARISION

Circuit	# Samples	Moment Order	PEM Error(%)	MAXENT Error(%)
SRAM	200	6	46.349	11.85
		8	30.656	3.988
		10	15.577	3.281
		12	9.4457	3.394
		14	6.6038	3.181
		18	198.97	5.470
Op. Amp.	200	10	125.54	30.943
		12	116.39	30.881
		14	108.43	5.374
		16	102.05	5.506
		18	93.793	5.567
		20	111.49	5.584

TABLE II
ACCURACY COMPARISON

Circuit	# Samples	Moment Order	PEM Error(%)	MAXENT Error(%)
SRAM	300	6	46.117	11.043
		8	30.251	5.331
		10	15.097	6.046
		12	11.341	5.818
		14	10.74	6.516
		18	200	6.222
Op. Amp.	800	10	126.51	28.271
		12	117.26	3.851
		14	108.40	4.232
		16	101.110	3.679
		18	94.682	3.465
		20	89.264	3.568

IV. Piecewise Distribution Model

In the previous section, we saw that MAXENT is a robust method for statistical circuit performance modeling. It guarantees stability for monotonic moments and offers high accuracy compared to other statistical modeling algorithms. However, we note that MAXENT is a global moment matching approach which offers high accuracy in the bulk of the distribution, but is unlikely to capture the accuracy in the tail (high sigma) region of the distribution. To this end, MAXENT is an insufficient approach when modeling the high sigma behavior of circuit performance distributions.

In this section, we propose PDM (Piecewise Distribution Model) to accurately and effectively model the high sigma portion of non-linear distributions from circuits in high dimensionality. The motivation behind PDM is to accurately model the tail distribution of circuit behavior by using region specific moments. In general, moment matching techniques such as [1], [26] use moments that may accurately reflect the bulk or body of the distribution. However, these global approximation methods use general probabilistic moments which give very little information about the high sigma areas and thus fail to accurately model the tail distribution. To this end, PDM utilizes moment matching to approximate the high sigma distribution by using *region specific moments* which capture highly accurate information in regions of interest. In general, an *arbitrary* number of segments can be used to model the overall distribution. Without losing generality, we break the total distribution into two segments - the first distribution (Segment1) matches the low sigma region and is accurate in the body (typically $\leq 4\sigma$) while the second distribution (Segment2) matches the high sigma region and is accurate in the tail (typically $\geq 4\sigma$). The flow of the method is shown in Fig. 7 while details are given below.

A. Building the Segment1 Distribution

To build the Segment1 distribution, we first draw samples $q_i; i = \{1, ..., N_1\}$ from input parameter distributions $f(x_j); j = \{1, ..., p\}$ where p is the number of variables. Next, we simulate these samples using a circuit simulator to obtain circuit behavior outputs $y_i; i = \{1, ..., N_1\}$. Finally, sample probabilistic moments μ_k are calculated and matched using MAXENT as outlined in [26], [33]. Depending on the number of moments that are matched, we will obtain different Segment1 distributions. However, the exact number of moments to be matched is unknown because we do not know which set of moments map to different areas of the distribution [27]. Consequently, we sweep across a range of values $k = 5, 7, 9, ..., K$ to build multiple Segment1 distributions and select a single, "optimal" Segment1 distribution as explained below.

B. Selecting the Optimal Segment1 Distribution

One of the key characteristics of non-Gaussian distributions is that the gradient of their CDFs are monotonically increasing, i.e. the change in circuit behavior for a fixed change in probability continuously increases as the sigma value increases. Here, the sigma value is simply the standard Z-score of a Standard Normal distribution, $P(Z \geq \sigma)$. On the other hand, the gradient is constant for a Gaussian distribution. This is illustrated in Fig. 8 which shows the gradient of the CDF for a LogNormal (non-Gaussian) distribution vs a Gaussian distribution. Here, although the LogNormal distribution is a mathematical distribution, we label the $y - axis$ of the figure as *Circuit Behavior* to emphasize that this type of circuit behavior is of interest. Consequently, we select the optimal Segment1 distribution by choosing the one with a monotonically increasing gradient.

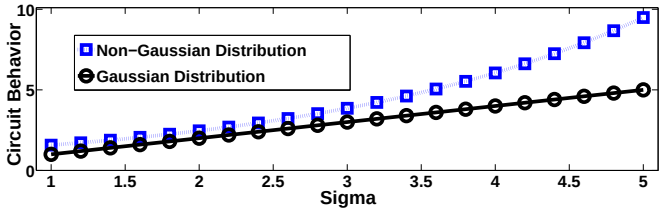


Fig. 8. Slope of Gaussian vs Non-Gaussian Distribution

In order to gauge the monotonicity of the gradient, we turn to Spearman's rank correlation coefficient [37]. Unlike the conventional Pearon correlation coefficient, which directly measures the correlation between two sets of variables, we utilize Spearman's rank correlation coefficient because it measures the *monotonic* relationship between two sets of variables. Specifically, the correlation coefficient ρ is a measure of how well a set of data can be described using a monotonic function. A coefficient of +1 indicates strong correlation to a monotonically increasing function while a coefficient of -1 indicates strong correlation to a monotonically decreasing function. To this end, we measure the gradient of the CDF for various body distributions and compare the data set to a monotonically increasing set using Spearman's Coefficient ρ and select the distribution with the largest, positive coefficient. Fig. 9 compares various Segment1 distributions, each built with a different number of moments, that are used for approximating a non-Gaussian distribution. We see that the coefficient for 5 of 6 distributions indicates that the gradient data set is monotonically decreasing or uncorrelated. However, there is a single distribution using 14 moments with a coefficient of $\rho = 0.98$, indicating it is a monotonically increasing set and should be used as the optimal Segment 1 distribution. In general, the optimal Segment 1 distribution may not have 14 moments.

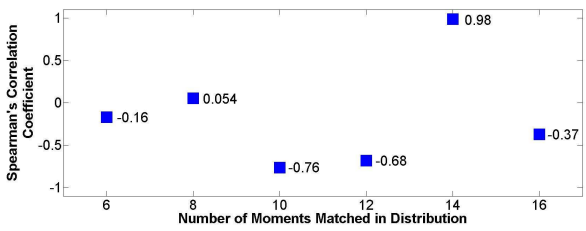


Fig. 9. Spearman's Correlation of Distributions with Different Moments

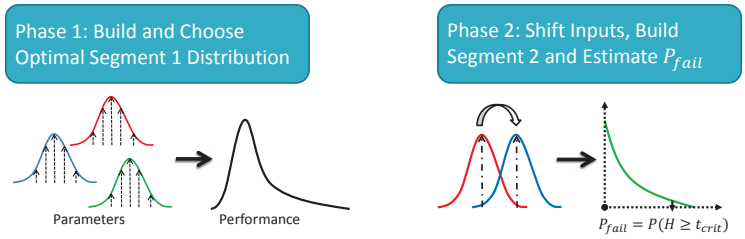


Fig. 7. PDM contains 2 Phases: building the Segment1 distribution and selecting the optimal Segment1 distribution; shifting input parameters to build the Segment2 distribution, and estimating the final probability

To confirm that this is the optimal choice of the above example, we compare the estimated data from the selected Segment1 distribution (strong Spearman's correlation), one non-selected distribution (poor Spearman's correlation), and the ground truth values as shown in Fig. 10. We see that the selected distribution matches very well with the ground truth because both distributions are non-Gaussian and exhibit monotonically increasing gradients. On the other hand, the distribution with poor correlation is very inaccurate. We utilize this combination of gradient and Spearman's correlation to select the optimal Segment1 distribution used in PDM.

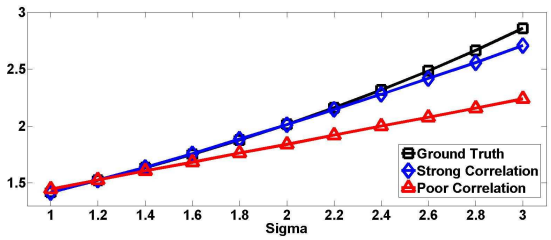


Fig. 10. Segment1 Comparison using Spearman's Correlation Results

C. Shifting Input Distributions and Building the Segment2 Distribution

The motivation behind shifting the input distributions is to draw more samples that yield an output in the tail of the original circuit behavior distribution. By generating more samples in this region, we can generate region specific moments that are highly accurate in the tail. To obtain moments ν_l that are specific to the tail of the distribution, we must shift the mean of the input parameter distributions from m to $\hat{m}$ for each input parameter individually. To shift the mean, we first find the largest circuit behavior y_{max} from the set y_i used when building the Segment1 distribution. The value of y_{max} is directly impacted by the sampling algorithm and number of samples in N_1. Finding the optimal y_{max} is out of the scope of this paper, and the y_{max} used in the proposed application attempts a shift towards the general vicinity of parameter samples that produce tail-like circuit behaviors. Each circuit behavior y_i has a corresponding set of input samples q_j for each input parameter $j = 1, ..., p$. The largest circuit behavior y_{max} will have a sample value q_j^* for each input parameter $j = 1, ..., p$. To obtain the shifted distributions, we simply shift the mean m_j of parameter j to the sample q_j^*.

Once the input parameters are shifted, an additional N_2 samples $\hat{q}_i; i = 1, ..., N_2$ are drawn and simulated yielding an output $\hat{y}_i; i = 1, ..., N_2$. To ensure that the moments ν_l are comprised of information *only* in the tail distribution, we must first screen the simulated data $\hat{y}_i$ such that only samples that lay in the tail are used. To do this, we simply pick a circuit behavior t^* that separates the Segment1 distribution and the next distribution, in this case Segment2. The value of t^* is obtained by selecting a sigma point s in the Segment1 distribution and extracting the corresponding circuit behavior. Typically, s is chosen to be a sigma value between 3 and 4 as this is where the long, flat region of the tail begins as shown in Fig. 8. Next, the circuit behavior values are screened to obtain $w_k = \hat{y}_i \geq t^*; k = 1, ..., N_3$ where N_3 is the number of points beyond t^*. Because the output was screened, we ensure that the moments ν_l shall only be reflective of the tail distribution's domain and not be polluted by information outside of it.

Finally, to build the Segment2 distribution, we calculate $l = 4$ moments using $\mu_i = \int x^i p(x) dx$ and match them using maximum entropy as in [26], [31], [34], [33]. The motivation behind using only 4 moments is that this forces the maximum entropy method to yield an exponential distribution as shown in [36]. The exponential distribution is a good approximation of the tail as it is monotonically decreasing and can easily be obtained using the maximum entropy method.

D. Reweighing Segment2 via Conditional Probability

Once the Segment2 distribution is obtained, the probability for a specified circuit behavior t_{spec} can be obtained; however, it will be inherently biased because the input parameters were shifted to draw more important samples. To resolve this issue, we use conditional probability to "re-weigh" probabilities as follows

$$P(H \geq t_{spec}) = P(H \geq t_{spec} | B \geq t^*) * P(B \geq t^*) \quad (16)$$

Where H is the random variable associated with the Segment2 distribution, B is the random variable associated with the Segment1 distribution, t_{spec} is the circuit behavior whose probability is of interest, and t^* is the circuit behavior for sigma point s. The conditional probability relationship in (16) works well when the two distributions H and B are identical, i.e. if we are calculating conditional probability under one distribution, or if they share the same mean. However, this equation does not hold true in the proposed algorithm. This is demonstrated by rearranging (16) as shown in (17)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2562923, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

$$P(B \geq t^*) = \frac{P(H \geq t_{spec})}{P(H \geq t_{spec}|B \geq t^*)} \quad (17)$$

For a new point $t^{'}_{spec}$, the relationship is

$$P(H \geq t^{'}_{spec}) = P(H \geq t^{'}_{spec}|B \geq t^*)P(B \geq t^*) \quad (18)$$

$$P(B \geq t) = \frac{P(H \geq t^{'}_{spec})}{P(H \geq t^{'}_{spec}|B \geq t^*)} \quad (19)$$

Rearranging (17) and (19) and equating the common term yields

$$P(B \geq t) = \frac{P(H \geq t_{spec})}{P(H \geq t_{spec}|B \geq t^*)} = \frac{P(H \geq t^{'}_{spec})}{P(H \geq t^{'}_{spec}|B \geq t^*)} \quad (20)$$

Clearly this relationship holds perfectly when the distributions from the numerator and denominator (joint and conditional, respectively) are identical as in importance sampling algorithms such as [17]. However, because PDM performs the re-weighing process in the output domain, the modeled tail and the true distribution may be shaped extremely differently. In other words, because the B and H distributions are necessarily two different random variables, the relationship in (16) must be modified to account for the shape mismatch that inherently arises due to the unknown shape of the distributions. Consequently, we propose a dynamic scaling technique that additionally reweighs the probability under the Segment2 distribution by a scaling factor β. The scaling factor acts as a heuristic correction factor that is calculated based on the indicator function of the subset w_k of the entire circuit behavior space, and the total number of outputs N_3 as shown in (22). Each approximation of different t_{spec} values has a different value of beta due to different values of the indicator function (21).

$$I(w_k) = \begin{cases} 0 \text{ if } w_k < t_{spec} \\ 1 \text{ if } w_k \geq t_{spec} \end{cases} \quad (21)$$

$$\beta = \sum_{k=1}^{N_3} \frac{I(w_k)}{N_3} \quad (22)$$

Using this scaling factor yields the final probability of a specified circuit behavior t_{spec} as (23)

$$P(H \geq t_{spec}) = P(H \geq t_{spec}|B \geq t^*) * P(B \geq t^*) * \beta \quad (23)$$

Fig. 11 shows an example of the difference in shape between the true tail distribution, the unscaled Segment2 distribution and the scaled Segment2 distribution. Additionally, we note that both Segment1 and Segment2 distributions are guaranteed to be stable, i.e. they will have a non-negative probability and therefore the CDF is guaranteed to be monotonic. This naturally arises because both distributions are calculated using the maximum entropy method and all moments in both segments are monotonically increasing.

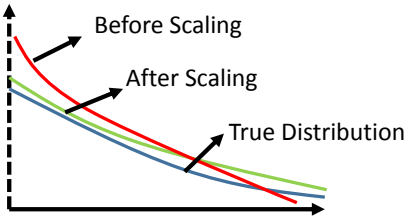


Fig. 11. Shape Issue in Conditional Probability

V. Experiment Results

A. Experiment Settings

We implemented PDM in MATLAB using simulation outputs from HSPICE. PDM is compared with Monte Carlo, moment matching algorithm MAXENT [26], High Dimensional Importance Sampling (HDIS) [17], and subset simulation (SUS) [25] to demonstrate that it offers significant speedup while maintaining higher accuracy than other methodologies that are targeted towards modeling the high sigma behavior of circuits.

The algorithm was tested against the mathematically known LogNormal distribution, along with the high sigma delay of a six stage clock path circuit and gain of an Operational Amplifier. The results show the estimated sigma for multiple t_{spec} values and are compared to Monte Carlo as ground truth. The Monte Carlo results were generated with roughly $8E6$ samples for the Time Critical Path and $2.5E6$ samples for the Operational Amplifier. Additionally, we compare the results to the MAXENT algorithm to show the improvements using a piecewise distribution model rather than a global approach. We also compare the results to HDIS to show that the re-weighing portion of PDM is accurate and robust for high dimensional circuits because it is independent of dimensionality. The independence is due to the re-weighing process occurring in the output domain where there is only a single variable. The source code of SUS is also obtained from its original authors for cross evaluation. Table III gives an overview of the variables used in each circuit.

TABLE III
Parameters of MOSFETs

Variable Name	Time Critical Path	OpAmp
Flat-band Voltage	†	
Threshold Voltage		†
Gate Oxide Thickness	†	†
Mobility	†	†
Doping concentration at depletion	†	
Channel-length offset	†	†
Channel-width offset	†	
Source/drain sheet resistance	†	†
Source-gate overlap unit capacitance	†	†
Drain-gate overlap unit capacitance	†	†

The time critical path circuit has six stages and nine process parameters per transistor for a total of 54 variables, while the circuit behavior of interest is the delay from input to output. Fig. 4 displays a schematic of the two-stage differential cascode operational amplifier, and is the same circuit as in [26]. The circuit has a total of thirteen transistors and four gain boosting amplifiers. In total, only ten transistors are considered

to be independently varied. However, transistors in the gain boosting amplifiers are also varied, though due to the mirrored properties of the circuit they are varied simultaneously and are counted as one variation. As such, although each transistor has seven process parameters resulting in a total of 70 variables, the true number of variables is much higher. In the proposed algorithm, the circuit behavior of interest is the gain $\frac{V_{out1}}{V_{in1}}$.

B. Experiment on Mathematical Distribution

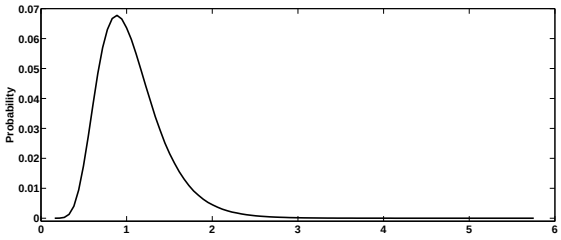


Fig. 12. LogNormal PDF

To illustrate the capability of modeling strongly non-Gaussian distributions, we use PDM to model a LogNormal distribution. The LogNormal distribution with mean and sigma parameters $\mu = 0$, $\sigma = 0.35$ was selected because of its strongly non-Gaussian behavior. A plot of the PDF of this distribution is presented in Fig. 12. The distribution appears to be Gaussian for a small portion due to the bell shaped curve, but it has a very long tail, giving it the non-Gaussian properties that are of interest.

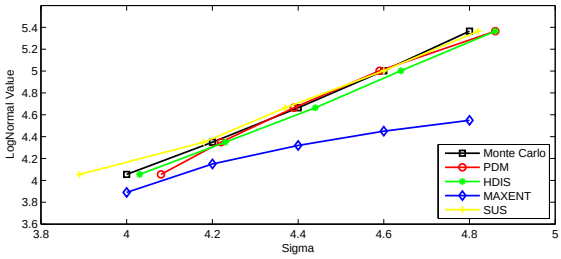


Fig. 13. LogNormal Sigma Behavior

Fig. 13 shows the high sigma modeling results for Monte Carlo, PDM, HDIS, MAXENT, and SUS at multiple t_{spec} points. The figure is the CDF zoomed into the tail area with the x-axis as sigma and y-axis as the value of the random variable, precisely circuit behavior. Here sigma is used to represent probability, i.e. $4\sigma \approx 0.000064$ in the tail. The motivation for this type of plot is to best represent the non-linear behavior of a non-Gaussian PDF. Additionally, it shows only the high sigma behavior rather than the overall distribution because that is the motivation and focus behind this algorithm.

While the number of samples required for SUS ranges from 5800 to 7400 in the experiment setup, HDIS, MAXENT and PDM each used a total of 4000 samples, with PDM using 3000 samples to calculate the Segment1 distribution and 1000 samples to calculate the Segment2 distribution. In this case, the point s that separates Segment1 and Segment2 is selected to be the 4 sigma point, i.e. whatever circuit behavior that corresponds to a tail probability of $6.4E-5$ in the Segment1 distribution. By introducing the Segment2 distribution at the point s, PDM is able to avoid any errors that MAXENT suffers from, allowing PDM to match almost identically with the Monte Carlo results up to 4.8 sigma. By utilizing region specific moments and doing a piecewise approximation of the distribution, PDM keeps consistently small errors. On the other hand, the MAXENT algorithm begins to lose accuracy and fails to capture the tail of the distribution because it only uses one distribution to model the overall behavior.

Furthermore, we see that all the high sigma modeling methods, HDIS, SUS, and PDM have accuracy comparable to Monte Carlo (with less than 0.1*sigma deviation). We can observe some deviations from Monte Carlo on PDM and SUS, but as statistical algorithms, those deviations are expected and are within tolerance as they are small and unsystematic. Table IV shows the error in estimated sigma for PDM. The error is between -0.25% and 2% all the way to the 4.8 sigma point.

We also note that MAXENT and PDM do *not* assume the distributions to be matched are Gaussian distributions because they do not match only 3 moments. [36] outlines that the maximum entropy moment matching method can be forced to assume a Gaussian distribution if we match exactly 3 moments. However, because we sweep through a wide range of moments for both MAXENT and PDM, we, in general, will never pick a Gaussian distribution because it does not agree with the gradient criteria selected by Spearman's correlation coefficient. Consequently, the high error that MAXENT suffers from is due to its limitation of using one set of moments, not from any assumptions about its model.

TABLE IV
SIGMA ERROR FOR LOGNORMAL

True Sigma	Estimated Sigma	% Error
4.0	4.0786	1.9650%
4.2	4.2224	0.5333%
4.4	4.3886	-0.2591%
4.6	4.5888	-0.2435%
4.8	4.8569	1.1854%

C. Experiments on Circuits

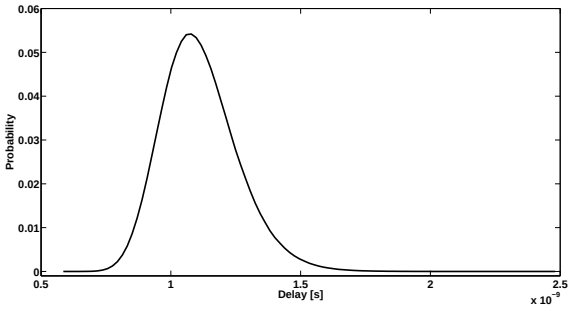


Fig. 14. Clock Path PDF

The Monte Carlo distribution of the time critical path circuit delay is presented in Fig. 14. Because the circuit operates at a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2562923, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

very low VDD level, it behaves in a slightly non-linear way. The distribution, while not as long tailed as the LogNormal, has a more elongated tail than a Gaussian distribution.

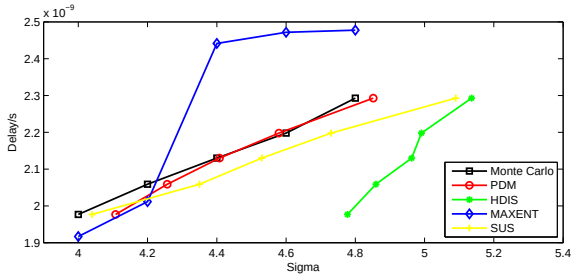


Fig. 15. Clock Path Sigma Behavior

Fig. 15 shows the high sigma modeling results for Monte Carlo, PDM, HDIS, MAXENT, and SUS at multiple t_{spec} points. HDIS, MAXENT and PDM each used a total of 4000 samples, with PDM using 3000 samples to calculate the Segment1 distribution and 1000 samples to calculate the Segment2 distribution. Once again in PDM, the point s is the 4 sigma point from the Segment1 distribution. By introducing the Segment2 distribution at the point s, PDM is able to avoid any errors that MAXENT may suffer from. This is most apparent at the 4.4 sigma point and beyond. Additionally, PDM is able to capture the increase in slope as the circuit approaches higher sigma. On the other hand, MAXENT is able to perform somewhat well up to 4.2 sigma but then blows up and becomes completely inaccurate afterwards. The significant increase in accuracy with PDM is, again, due to matching region specific moments that allow piecewise approximation of the distribution. Because MAXENT uses a single distribution to make a global approximation it is unable to capture the tail of the distribution and instead models the high sigma points purely as noise. We again note that MAXENT does not assume the distribution is a Gaussian model, so its error is due to limitations of using one set of moments to model the total distribution which PDM does not suffer from.

SUS uses between 5803 and 9010 samples for different sigma points, which is slightly less efficient compared to PDM (4000 samples). In terms of accuracy, the probability estimated by SUS follows the same trend of MC and PDM, i.e., the curve from SUS is almost parallel to the curves of MC and PDM as illustrated in Fig. 15. However, if we compare the probability estimated by PDM and SUS in Fig. 15 in detail, we can find that at lower threshold (1.98ns), SUS has the smallest deviation (about 0.04*sigma at 4 sigma) from MC. As we move towards higher thresholds, and thus allow less failure samples, we find that the deviation between the estimated probability of MC and SUS grows constantly, i.e. 0.13-0.15*sigma at 4.2-4.6 sigma, and finally 0.29*sigma difference at 4.8 sigma. SUS does not experience such notable estimation error on the 1-dimensional lognormal distribution. However, at high dimensions, samples generated by only a few Markov chains could be insufficient to cover the entire failure region(s), leading to large deviation compared with MC. Moreover, as samples in each phase of SUS are generated with a modified Metropolis (MM) algorithm using the previously failed samples as the seed, the estimation error in one phase may accumulate and propagate to the next. When the problem scales to higher sigma values, SUS would require an increased number of phases to better cover the smaller failure region, which may lead to an unwanted runtime overhead. On the other hand, a smaller number of phases results in larger deviation from the MC estimate as was mentioned above. Compared with SUS, PDM results stay closer to MC with less than 0.1*sigma deviation.

Furthermore, we see that the results from HDIS are completely inaccurate compared to both Monte Carlo and PDM. HDIS is unable to come anywhere near the proper sigma value for any of the points that it estimates. This is likely inaccurate from a combination of high dimensionality and an inaccurate shift in the mean and sigma of the new sampling distribution that causes the re-weighing process to again become inaccurate. Simply put, if the shifting method is inaccurate the results from HDIS will be inaccurate. If a larger number of samples is used, then the shift and corresponding samples drawn from the new distribution will be more accurate; however, due to the run time prohibitive nature of high dimensional circuits, it is imperative to minimize the number of samples. On the other hand, the shifting method in PDM is more robust because the re-weighing process is performed in the output domain and is performed using conditional probability rather than as a ratio of two distributions. Table V shows the error in sigma between PDM and the ground truth from Monte Carlo. We see a worst case error of 2.7% at 4 sigma but significantly less errors at higher sigma values.

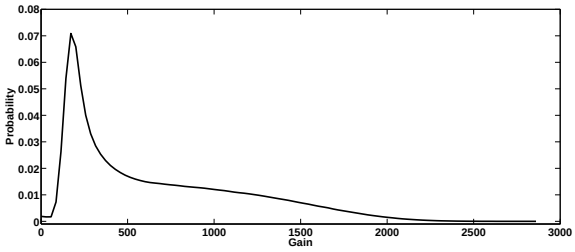


Fig. 16. Op. Amp PDF

The Monte Carlo distribution of the Operational Amplifier circuit gain is shown in Fig. 16. The distribution is heavily skewed and has a very sharp peak near the beginning and proceeds to drop very quickly, However, it also has a slightly flatter portion that eventually decreases to a long, flat region of the tail. It clearly has a long tail and behaves in a strongly non-Gaussian way.

Fig. 17 shows the high sigma modeling results for Monte Carlo, MAXENT, and PDM at multiple t_{spec} points. The figure shows only the high sigma behavior rather than the overall distribution because that is the motivation and focus behind this algorithm. Both MAXENT and PDM used a total of 3000 samples, with PDM using 2000 samples to calculate the Segment1 distribution and 1000 samples to calculate the Segment2 distribution. In the case of the OpAmp, the point s was determined to be the 3.6 sigma point rather than the 4 sigma point as in the previous cases due to the extremely long-tailed nature of the distribution. Before the point s, it's clear

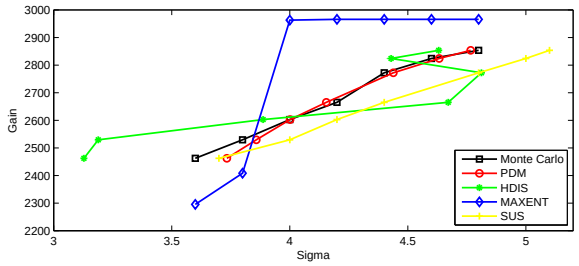


Fig. 17. Op Amp Sigma Behavior

that PDM has a larger error (roughly 5%) than in previous cases. However, when we introduce the Segment2 distribution, PDM is able to immediately recover and match the 3.8 sigma point closely and continues to match larger sigma points and the overall shape of the Monte Carlo curve very well. By introducing this second "piece" to model the distribution, we are able to get a significant increase in accuracy. On the other hand, the MAXENT method has a large error, blows up and returns noise values because it is unable to capture the tail of the distribution as it does not use moments that are specific to that region. We again note that MAXENT does not assume the distribution is a Gaussian model because it matches more than 3 moments. Hence, its error is due to limitations of using one set of moments to model the total distribution.

The SUS algorithm used between 5004 and 8216 samples at different sigma points. While SUS is able to capture the overall trend and shape of the Monte Carlo results, the sigma estimated by SUS is pessimistic with respect to Monte Carlo and tends to slightly overestimate the true sigma value, with small pessimism at lower sigmas (3.7 vs 3.6) and higher pessimism at larger sigmas (5.1 vs 4.8). The results observed on this circuit are similar to those on the clock path circuit, indicating a growing deviation between MC at higher sigmas.

We see that the results from HDIS are inaccurate and at one point has a huge jump in its results and is simply noisy throughout. Although the Operational Amplifier circuit is not as high dimensional as the Clock Path, HDIS is still unable to properly model the high sigma region. Again, the inaccuracy is most likely from an inaccurate shift in the mean and sigma of the new sampling distribution that causes the re-weighing process to again become inaccurate. Table V shows the error in estimated sigma between PDM and the ground truth from Monte Carlo. We see very accurate results with a worst case error of about -1% at 4.2 sigma.

TABLE V
SIGMA ERROR FOR CIRCUITS

Time Critical Path			Op Amp		
True Sigma	Estimated Sigma	% Error	True Sigma	Estimated Sigma	% Error
4.0	4.1077	2.693%	4.0	4.0015	0.0375%
4.2	4.2571	1.360%	4.2	4.1547	-1.0786%
4.4	4.4080	0.182%	4.4	4.4386	0.8773%
4.6	4.5793	-0.450%	4.6	4.6329	0.7152%
4.8	4.8517	1.077%	4.8	4.7662	-0.7042%

D. Speedup Comparison

To analyze the efficiency of the proposed method, we compare the number of samples required by PDM to the number of samples used for Monte Carlo. Since the LogNormal distribution is a mathematically known circuit and requires no Monte Carlo simulations, we exclude that speedup comparison. In the clock path circuit, PDM requires a total of 4000 samples - 3000 samples for the body distribution and 1000 for the hybrid distribution. In the Operational Amplifier, PDM requires a total of 3000 samples - 2000 samples for the Segment1 distribution and 1000 for the Segment2 distribution. Table VI compares the Monte Carlo and PDM runtime requirements and the speedup for all circuit examples. We note that the speedup of the algorithm compared to Monte Carlo will vary based on the number of samples that are used; however, it is clear that PDM offers a significant speedup at very little loss in accuracy.

TABLE VI
SPEEDUP COMPARISON

Circuit	Monte Carlo Runtime	PDM Runtime	Speedup
Clock Path	8,000,000	4000	$2000x$
Op. Amp.	2,500,000	3000	$833x$

VI. CONCLUSION

In this paper, we presented two novel algorithms for statistical performance modeling of circuits. The first algorithm was based on the maximum entropy moment matching method which was originally proposed in the communications and signal processing field. The MAXENT algorithm is provably stable under general statistical circuit analysis methods. Experimental results indicate that it offers high accuracy and stability when compared to other moment matching methods [1], [3]. However, MAXENT is unable to accurately model the high sigma behavior of non-Gaussian circuits and is therefore unsuitable for yield analysis. To this end, we proposed PDM - a piecewise distribution model that performs region based moment matching to extract the PDF of circuit performance. PDM is provably stable because it is based on the maximum entropy method. Furthermore, it is able to model the high sigma regions of the circuit performance PDF. In particular, we introduced a second distribution based on a set of moments that are accurate in the tail of the PDF leads to significantly improved accuracy over MAXENT [26] with little error compared to Monte Carlo.

We demonstrated that PDM performs as well or better than other state-of-the-art statistical modeling methodologies [26], [17], [25]. The importance sampling technique in [17] is inaccurate for high-dimensional circuits due to the "curse-of-dimensionality" [21] from the re-weighing procedure applied to every input parameter that is shifted. While PDM employs a similar input parameter shifting, its re-weighing procedure is performed only in the *output* domain which is one-dimensional and thus avoids issues with the degeneration and unbounded distribution support. Consequently, it is able to efficiently handle high-dimensional cases without instability. Finally, SUS offers higher accuracy than both MAXENT and Importance

Sampling, especially in the lower-dimensional experiments. The estimates from SUS match the general trend of those from MC and PDM, and often provides the very high accuracy at lower sigma values. However, at higher sigma values SUS suffers from some inaccuracy on both high dimensional circuit examples, on which the failure regions are difficult to capture with only a few Markov chains. Furthermore, because SUS generates samples in each phase using the previously failed samples as the seed, estimation errors in one phase may accumulate to the latter phases.

In the future, we plan to develop a weighted moment matching based approach that allows us to pick and choose the important moments of a distribution. The motivation behind this is not all moments are important to the distribution, e.g. in a Gaussian distribution only even order moments are non-zero, and therefore applying more weight to "important" moments may help improve accuracy and reduce noise.

VII. Acknowledgement

We would like to express our sincere thanks to Prof. Xin Li and Dr. Shupeng Sun for sharing the source code of SUS for evaluation.

References

[1] F. Gong, H. Yu, and L. He, "Stochastic analog circuit behavior modeling by point estimation method," *Proceedings of the 2011 international symposium on Physical design*, pp. 175–182, 2011.

[2] S. Nassif, "Modeling and analysis of manufacturing variations," *IEEE article on Custom Integrated Circuits*, pp. 223–228, 2001.

[3] X. Li, J. Le, P. Gopalakrishnan, and L. Pileggi, "Asymptotic probability extraction for non-normal distributions of circuit performance," *Proceedings of the 2004 IEEE/ACM International article on Computer-aided design*, pp. 2–9, 2004.

[4] S. Wang, G. Leung, A. Pan, C. O. Chui, and P. Gupta, "Evaluation of digital circuit-level variability in inversion-mode and junctionless finfet technologies," *Electron Devices, IEEE Transactions on*, vol. 60, no. 7, pp. 2186–2193, 2013.

[5] S. Wang, H. Lee, F. Ebrahimi, P. K. Amiri, K. L. Wang, and P. Gupta, "Comparative evaluation of spin-transfer-torque and magnetoelectric random access memory," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, 2016.

[6] H. Kaul, M. Anders, S. Hsu, A. Agarwal, R. Krishnamurthy, and S. Borkar, "Near-threshold voltage (ntv) design: opportunities and challenges," in *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012, pp. 1153–1158.

[7] S. Vrudhula, J. Wang, and P. Ghanta, "Hermite polynomial based interconnect analysis in the presence of process variations," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 10, pp. 2001–2011, 2006.

[8] C. Jacoboni and P. Lugli, *The Monte Carlo method for semiconductor device simulation*. Springer, 2002, vol. 3.

[9] W. Wu, Y. Shan, X. Chen, Y. Wang, and H. Yang, "Fpga accelerated parallel sparse matrix factorization for circuit simulations," *Reconfigurable Computing: Architectures, Tools and Applications*, pp. 302–315, 2011.

[10] X. Chen, W. Wu, Y. Wang, H. Yu, and H. Yang, "An escheduler-based data dependence analysis and task scheduling for parallel circuit simulation," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, no. 99, pp. 1–5, 2011.

[11] W. Wu, F. Gong, H. Yu, and L. He, "Exploiting parallelism by data dependency elimination: A case study of circuit simulation algorithms," 2013.

[12] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of sram designs in the presence of rare failure events," in *Proceedings of the 43rd annual Design Automation Conference*. ACM, 2006, pp. 69–72.

[13] K. Katayama, S. Hagiwara, H. Tsutsui, H. Ochi, and T. Sato, "Sequential importance sampling for low-probability and high-dimensional sram yield analysis," in *Proceedings of the International Conference on Computer-Aided Design*. IEEE Press, 2010, pp. 703–708.

[14] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan, "Breaking the simulation barrier: Sram evaluation through norm minimization," in *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*. IEEE Press, 2008, pp. 322–329.

[15] M. Qazi, M. Tikekar, L. Dolecek, D. Shah, and A. Chandrakasan, "Loop flattening & spherical sampling: highly efficient model reduction techniques for sram yield analysis," in *Proceedings of the Conference on Design, Automation and Test in Europe*. European Design and Automation Association, 2010, pp. 801–806.

[16] F. Gong, S. Basir-Kazeruni, L. Dolecek, and L. He, "A fast estimation of sram failure rate using probability collectives," *ACM International Symposium on Physical Design*, pp. 41–47, 2012.

[17] W. Wu, F. Gong, C. Gengsheng, and L. He, "A fast and provably bounded failure analysis of memory circuits in high dimensions," in *Proceedings of the 2013 Asia and South Pacific Design Automation Conference*. IEEE, 2013.

[18] W. Wu, S. Bodapati, and L. He, "Hyperspherical clustering and sampling for rare event analysis with multiple failure region coverage," in *Proceedings of the 2016 on International Symposium on Physical Design*. ACM, 2016, pp. 153–160.

[19] A. Singhee and R. A. Rutenbar, "Statistical blockade: very fast statistical simulation and modeling of rare circuit events and its application to memory design," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 8, pp. 1176–1189, 2009.

[20] W. Wu, W. Xu, R. Krishnan, Y.-L. Chen, and L. He, "Rescope: High-dimensional statistical circuit simulation towards full failure region coverage," in *Proceedings of the 51st Annual Design Automation Conference*. ACM, 2014, pp. 1–6.

[21] T. Bengtsson, P. Bickel, and B. Li, "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," *Probability and statistics: Essays in honor of David A. Freedman*, vol. 2, pp. 316–334, 2008.

[22] R. Y. Rubinstein and P. W. Glynn, "How to deal with the curse of dimensionality of likelihood ratios in monte carlo simulation," *Stochastic Models*, vol. 25, no. 4, pp. 547–568, 2009.

[23] A. Singhee, J. Wang, B. H. Calhoun, R. Rutenbar *et al.*, "Recursive statistical blockade: An enhanced technique for rare event simulation with application to SRAM circuit design," in *VLSI Design, 2008. VLSID 2008. 21st International Conference on*. IEEE, 2008, pp. 131–136.

[24] S. Sun, X. Li, H. Liu, K. Luo, and B. Gu, "Fast statistical analysis of rare circuit failure events via scaled-sigma sampling for high-dimensional variation space," in *Proceedings of the International Conference on Computer-Aided Design*. IEEE Press, 2013, pp. 478–485.

[25] S. Sun and X. Li, "Fast statistical analysis of rare circuit failure events via subset simulation in high-dimensional variation space," in *Computer-Aided Design (ICCAD), 2014 IEEE/ACM International Conference on*. IEEE, 2014, pp. 324–331.

[26] R. Krishnan, W. Wu, F. Gong, and L. He, "Stochastic behavioral modeling of analog/mixed-signal circuits by maximizing entropy," in *Quality Electronic Design (ISQED), 2013 14th International Symposium on*. IEEE, 2013, pp. 572–579.

[27] R. Durrett, *Probability: theory and examples*. Cambridge university press, 2010, vol. 3.

[28] H. Niederreiter, *Random number generation and quasi-Monte Carlo methods*. SIAM, 1992, vol. 63.

[29] E. Jaynes, "Information theory and statistical mechanics," *Physical review*, vol. 106, no. 4, p. 620, 1957.

[30] P. McCullagh, *Tensor methods in statistics*. Chapman and Hall London, 1987, vol. 161.

[31] X. Wu, "Calculation of maximum entropy densities with application to income distribution," *Journal of Econometrics*, vol. 115, no. 2, pp. 347–354, 2003.

[32] G. Judge and D. Miller, *Maximum entropy econometrics: Robust estimation with limited data*. John Wiley & Sons, 1997.

[33] L. Mead and N. Papanicolaou, "Maximum entropy in the problem of moments," *Journal of Mathematical Physics*, vol. 25, p. 2404, 1984.

[34] B. Chen, J. Hu, and Y. Zhu, "Computing maximum entropy densities: A hybrid approach," *Signal Processing: An International Journal (SPIJ)*, vol. 4, no. 2, p. 114, 2010.

[35] P. Feldmann and R. Freund, "Efficient linear circuit analysis by padé approximation via the lanczos process," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 14, no. 5, pp. 639–649, 1995.

[36] K. Conrad, "Probability distributions and maximum entropy," *Entropy*, vol. 6, no. 452, p. 10, 2004.

[37] M. G. Kendall, "Rank correlation methods." 1948.

Rahul Krishnan received the M.S and B.S. degrees in electrical engineering from the University of California, Los Angeles in 2014 and 2012, respectively. Currently, he works with Taiwan Semiconductor Manufacturing Company, Austin, Texas. His primary research interests are in statistical circuit modeling and simulation.

Wei Wu is a PhD graduate student in Electrical Engineering Department, UCLA since January 2012. Prior to that, he obtained his M.S. and B.S. degrees in electrical engineering from Beijing University of Aeronautics and Astronautics in 2010 and 2007, respectively. His research interests mainly focus on parallel and statistical circuit simulation. He also works on FPGA based hardware computing systems, such as FPGA based analog emulation platform, and FPGA based deep learning inference engine.

Srinivas Bodapati received the B.Tech. (honors) degree in electrical engineering from the Indian Institute of Technology, Kharagpur, India in 1995 and the MS and PhD degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign, in 2000 and 2003 respectively.

He was with Cadence Design Systems, Noida, India, from 1995 to 1997. From 1997 to 1998, he was with Texas Instruments, Bangalore, India. Since 2003, he has been at Intel Corporation, Santa Clara, CA, working on statistical methods for pre-silicon analysis and modeling of power, timing and circuit reliability. His research interests include high dimensional modeling and analysis, rare event prediction, machine learning algorithms in classification, clustering and anomaly detection

Lei He Dr. Lei He is a professor at electrical engineering department, University of California, Los Angeles (UCLA) and guest chair professor at Fudan University, Shanghai, China. He was a faculty member atUniversity of Wisconsin, Madison between 1999 and 2002, consulted Cadence Design Systems, Cisco, Empyrean Soft, Hewlett-Package, Intel, and Synopsys, and was a founding technical advisory board member for Apache Design Solutions and Rio Design Automation.

Dr. He obtained Ph.D. degree in computer science from UCLA in 1999. His research interests include internet of things for sustainability and home automation, and machine learning on a chip. He has published one book and over 200 technical papers with many best paper nominations and awards including the 2010 ACM Transactions on Electronic System Design Automation Best Paper Award and the 2011 IEEE Circuit and System Society Darlington Best Paper Award.