# Why Quasi-Monte Carlo is Better than Monte Carlo or Latin Hypercube Sampling for Statistical Circuit Analysis

Amith Singhee, *Member, IEEE,* and Rob A. Rutenbar, *Fellow, IEEE*

***Abstract*—At the nanoscale, no circuit parameters are truly deterministic; most quantities of practical interest present themselves as probability distributions. Thus, Monte Carlo techniques comprise the strategy of choice for statistical circuit analysis. There are many challenges in applying these techniques efficiently: circuit size, nonlinearity, simulation time, and required accuracy often conspire to make Monte Carlo analysis expensive and slow. Are we—the integrated circuit community—alone in facing such problems? As it turns out, the answer is "no." Problems in computational finance share many of these characteristics: high dimensionality, profound nonlinearity, stringent accuracy requirements, and expensive sample evaluation. We perform a detailed experimental study of how one celebrated technique from that domain—quasi-Monte Carlo (QMC) simulation—can be adapted effectively for fast statistical circuit analysis. In contrast to traditional pseudorandom Monte Carlo sampling, QMC uses a (shorter) sequence of deterministically chosen sample points. We perform rigorous comparisons with both Monte Carlo and Latin hypercube sampling across a set of digital and analog circuits, in 90 and 45 nm technologies, varying in size from 30 to 400 devices. We consistently see superior performance from QMC, giving 2× to 8× speedup over conventional Monte Carlo for roughly 1% accuracy levels. We present rigorous theoretical arguments that support and explain this superior performance of QMC. The arguments also reveal insights regarding the (low) latent dimensionality of these circuit problems; for example, we observe that over half of the variance in our test circuits is from unidimensional behavior. This analysis provides quantitative support for recent enthusiasm in dimensionality reduction of circuit problems.**

***Index Terms*—Latin hypercube sampling (LHS), low-discrepancy sequence, Monte Carlo methods, quasi-Monte Carlo (QMC), statistical circuit analysis.**

## I. Introduction

DEVICE scaling continues to dramatically increase the relative impact of parametric variability on circuit performance. Circuit designers must now contend with these variations to ensure the robustness of their circuit designs. Traditional corner-based margining has given way to more accurate Monte Carlo simulation [1] for most transistor-level designs. In a few special cases, analytical methods exist to cast these inherently statistical problems into deterministic formulations, e.g., circuit tuning for combinational logic under statistical yield and timing constraints, as in [2], or statistical static timing analysis, as in [3]. Unfortunately, such analytical solutions remain rare and in the general case, some combination of complex statistics, high dimensionality, strong nonlinearity, or non-normality, stringent accuracy requirements, and expensive performance evaluation (e.g., SPICE simulation) thwart our analytical aspirations, leaving us with Monte Carlo simulation methods [4].

Over the years, Monte Carlo has become the standard technique for statistical simulation of circuits and for yield estimation during the design phase [1], [5]– [7]. The reason for this is that Monte Carlo is applicable to arbitrary circuits, arbitrary statistical models, and all performance metrics of interest, while allowing arbitrary accuracy. On the other hand, we gain the flexibility and accuracy of Monte Carlo at the cost of speed: a single Monte Carlo run can cost a few thousand SPICE simulations, and higher accuracy requirements demand longer runs.

There has been much research for methods to address these problems. One strategy is to replace Monte Carlo simulation altogether, e.g., by using acceptance region models [8]–[10]. However, these methods often do not scale well with dimensionality. Another strategy is to replace expensive SPICE simulations, e.g., by using response surface models [11]–[14]. However, these methods gain speed by sacrificing accuracy, and in many cases the resulting accuracy loss is unacceptable. An ideal solution would speed up Monte Carlo simulation *directly*, while still using SPICE simulations and maintaining the generality of its application. For example, [1] studies variance reduction techniques that exploit some problem characteristics to improve the accuracy (or reduce the simulation count) of Monte Carlo. [6] looks specifically at one widely used variance reduction method: *Latin hypercube sampling* (LHS); the paper presents some preliminary application of LHS to single transistor test cases, but does not present any rigorous analysis of the method or of its application to other types of circuits.

An alternative approach—and the one we focus on in this paper—speeds up Monte Carlo by improving the sample generator itself. In typical use, a pseudorandom generator

Manuscript received June 5, 2009; revised January 27, 2010 and May 14, 2010; accepted June 2, 2010. Date of current version October 20, 2010. This work was supported in part by the Semiconductor Research Corporation (SRC), and by C2S2, the Focus Center for Circuit and System Solutions, one of the five research centers funded under the Focus Center Research Program, an SRC program. This paper was recommended by Associate Editor M. Orshansky.

A. Singhee is with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: asinghe@us.ibm.com).

R. A. Rutenbar is with the University of Illinois at Urbana-Champaign, Urbana, IL 61820 USA (e-mail: rutenbar@illinois.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2010.2062750

[15] is employed to generate the Monte Carlo sample sequence. However, there exists a completely different class of sampling methods called low-discrepancy sequences (LDS) [16]. In contrast to standard pseudorandom sequences, these are deterministic sequences with no random component. The points in the sequence are generated to satisfy some rigorous notion of uniform coverage of the sampling space. Monte Carlo simulation that employs these deterministic sequences in place of (pseudo-)random sequences is termed as quasi-Monte Carlo (QMC).

Although the mathematical foundations for low-discrepancy sampling date back to the 1930s [17], it is interesting to note that QMC gained widespread visibility only recently, in the field of computational finance [18]. The analysis of many practical financial instruments, e.g., the price of a bond, is most naturally cast as a high-dimensional stochastic integration problem. In this arena, QMC has produced some spectacular successes. For example, in [19], researchers at IBM studied the pricing of a five-year discount bond, comprising a 1439-dimensional statistical integral. They observed a QMC speedup of about 150 for an accuracy level of one basis point (i.e., a relative accuracy of $10^{-4}$) when compared to random Monte Carlo. Results such as these motivated us to study the application of QMC to circuit analysis.

A preliminary version of this paper was presented in [20] (and spawned some follow-up work studying the use of QMC for statistical static timing analysis [21], [22]). In this paper, we go much further. We compare also against the popular LHS method, which can be cast as a randomized form of QMC, as discussed in [24]. We present a unifying theoretical and comparative analysis of the three methods that exposes insightful properties regarding the implicit structure of the chosen circuit problems—specifically their *latent dimensionality* [11]. This is the true underlying true dimensionality of the problem, more specifically the size of the smallest subset of (possibly transformed) parameters that largely explain the relevant circuit behavior. In [11], we studied this latent dimensionality from the viewpoint of performance modeling and showed that the true dimensionality of circuit behavior may be much lower than the number of circuit parameters. This observation has also been made by other researchers, such as [14]. In this paper, we explore these ideas in the context of statistical simulation. We use our analysis to show why QMC can often be a better choice compared to both Monte Carlo and LHS for circuit applications. A similar study is presented in [23] constrained to the case of timing analysis, which involves only max() and + operations. In this paper, although we do not look at specifically at these operations or timing analysis, we address general numerical circuit simulation which sees a wider variety of functional behaviors in the problems of interest. We see interesting variety in the underlying dimensionality of the problem for different circuits and clearly see its influence on the relative performance of QMC, LHS, and Monte Carlo.

The rest of this paper is organized as follows. In the next section, we review some convergence results for conventional Monte Carlo, and related mathematical concepts of uniformity. Section III presents the idea of LDS and QMC and Section IV gives the construction for one particular LDS, the *Sobol'* sequence. In Section V, we describe some relevant problems faced by QMC in high dimensions and use the concept of *effective dimension* to propose a strategy to minimize these problems for circuit analysis. We briefly review LHS in Section VI and argue why QMC can perform better than LHS when appropriately used. A strategy based on randomized QMC is proposed in Section VII for a measure of QMC accuracy that is consistent with the measure for Monte Carlo and LHS statistical accuracy. Section VIII presents our experimental setup, observed results, and detailed comparative analysis of these results. Finally, we conclude in Section IX.

## II. Monte Carlo: Some Convergence Results

Monte Carlo methods are typically used to solve some integration problem of the following canonical form:

$$Q = \int_{C^s} f(\mathbf{x})d\mathbf{x} \quad C^s = [0, 1]^s \tag{1}$$

where $C^s$ is the unit cube in $s$ dimensions, and $f$ is some integrable function. We assume that our integration domain is the unit cube and all necessary transformations have been incorporated into the function $f$. Monte Carlo provides us a randomized quadrature (numerical integration) method to approximately compute this integral. The Monte Carlo estimate is given as

$$Q_n = \frac{1}{n}\sum_{i=1}^{n} y_i \quad y_i = f(\mathbf{x}_i) \tag{2}$$

where $\mathbf{x}_i$ are $n$ independent and identically distributed random points drawn from the $s$-dimensional uniform distribution $\mathcal{U}[0, 1)^s$. Any $s$-dimensional problem with different variable ranges, arbitrary statistical distributions, arbitrary nonlinearity, and so on can always be transformed into the canonical integral form, by including these properties into the function $f$, without any loss of generality. Thus, any problem of our interest can be redefined over the $s$-dimensional unit cube, including parametric yield computation for circuits [24]. Given this, let us look the convergence properties of standard Monte Carlo.

### A. Monte Carlo Convergence

In (2), since $\mathbf{x}_i$, and hence $y_i$, are i.i.d., the Law of Large Numbers [25] says that the Monte Carlo estimate $Q_n$ converges almost surely to the exact integral $Q$ as the sample size $n$ is increased; that is

$$P\left(\lim_{n\to\infty} Q_n = Q\right) = 1. \tag{3}$$

If we run multiple $n$-point Monte Carlo runs, we will obtain a different random sample set $\{\mathbf{x}_i\}_1^n$, and, hence, a different estimate $Q_n$ each time. As a result, the integration error of Monte Carlo is probabilistic in nature, and is given by the following.

*Theorem 1:* If $f$ has finite variance

$$\sigma^2(f) = \int_{C^s} (f(\mathbf{x}) - Q)^2 d\mathbf{x} \tag{4}$$

the r.m.s. error (standard deviation) of Monte Carlo is

$$\sigma_{\mathrm{MC}} = \sqrt{E[(Q - Q_n)^2]} \to \frac{\sigma(f)}{\sqrt{n}} \text{ as } n \to \infty. \tag{5}$$

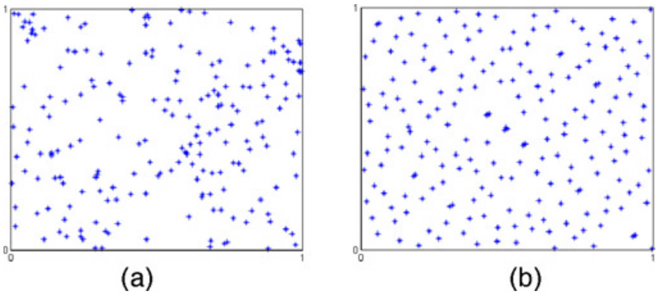

Fig. 1. In two dimensions, Sobol' points are more uniformly distributed than typical pseudorandom points. (a) 200-point pseudorandom point-set. (b) 200-point Sobol' pointset.

The proof follows immediately from the central limit theorem [25]. Hence, the Monte Carlo error decreases asymptotically as $n^{-1/2}$ for general integrable $f$. Variance reduction techniques [4] like importance sampling, LHS, stratified sampling, control variates, and Rao-Blackwellization, among others, typically target the proportionality constant ($\sigma(f)$) of this behavior. The QMC technique is complementary to these variance reduction techniques in the sense that it targets to speed up the $n^{-1/2}$ behavior and not the proportionality constant.

### B. *Discrepancy and Monte Carlo Integration Error*

Another way to study the error is using the concept of *discrepancy*. Fig.1 shows two sets of points in $C^2$ that might be used for Monte Carlo. In Fig. 1(a), we have 200 uniformly distributed pseudorandom points. The points are, indeed, uniformly distributed, but geometrically, they are not equally separated. We can see that the points exhibit both clumps and empty spaces. In Fig.1 we have a 200-point "deterministic" sample from the so-called Sobol' sequence. It is clear that the random sample is less geometrically uniform than the Sobol' sample. Discrepancy is a quantity used to reflect this geometric nonuniformity of points in a set.

There are several mathematical definitions of discrepancy [26], [27], one of the simplest being the *star discrepancy*, or the *$L_\infty$-discrepancy*. For any $n$-point sample in $C^s$, it is defined as

$$D_n^* = \sup_{J\subseteq C^s} \left|\frac{n_J}{n} - \mathrm{Vol}\,(J)\right| \qquad J = [\mathbf{0}, \mathbf{a}) : \mathbf{a} \in C^s \tag{6}$$

where $J$ is any $s$-dimensional hyperrectangle contained within $C^s$, with one corner at the origin $\mathbf{0}$, Vol($J$) is the volume of $J$, and $n_J$ is the number of points inside $J$. Hence, the star discrepancy $D_n^*$ measures how well the (relative) volume of any origin-anchored hyperrectangle in the unit cube is approximated by the fraction of points that lie in that volume. Surprisingly enough, pseudorandom samples from the standard uniform distribution $\mathbf{x}_i \sim \mathcal{U}[0, 1)^s$ may show extremely large discrepancy, as illustrated in Fig. 1(a).

The question now is, how is the error in the Monte Carlo integration related to this geometric placement or discrepancy of a point-set? The Koksma–Hlawka theorem provides an answer.

*Theorem 2 (Koksma–Hlawka [28], [29]):* If function $f$ has bounded variation in the sense of Hardy and Krause, then the Monte Carlo error is bounded as follows:

$$\epsilon(f) = |Q - Q_n| \le V(f) D_n^* \tag{7}$$

where $V(f)$ is the variation of $f$ in the sense of Hardy and Krause, and $D_n^*$ is the star discrepancy of the point-set. $V(f)$ is a measure of the total variation of the function over the unit cube, and has a rather technical definition that is not relevant to our discussion: we refer the reader to [30]. Here, we only illustrate the concept using the example of a smooth function in one dimension. For such a function

$$V(f) = \int_0^1 |df| \tag{8}$$

which is just the integral of the absolute value of the gradient of $f$. Hence, the more the function changes over the interval, higher is the value of $V(f)$. $V(f)$ for arbitrary, possibly nonsmooth, $f$ is similar in concept but more complex in definition [30].

Inequality (7) provides an upper bound on the integration error for Monte Carlo using any given point-set. It is particularly attractive because it separates out the two influences on the error: the properties of the function $f$, and the properties of the point-set. The larger implication is that sample points with *lower* discrepancy may produce integral estimates with *lower* errors. The first obvious question is, what is the discrepancy for standard Monte Carlo? For a random sequence of points uniformly distributed over $C^s$, it has been shown that [31]

$$D_n^* = O\left(\left[\frac{\log\log n}{n}\right]^{\frac{1}{2}}\right) \tag{9}$$

with probability 1. Combining this with the deterministic bound in (7) we see a match with the $n^{-0.5}$ convergence of the probabilistic error bound (5) for standard Monte Carlo. But the real question, from (7) is, are there point sequences that guarantee a lower discrepancy? The answer, as Fig. 1(b) suggests, is "yes."

## III. Low-Discrepancy Sequences and Quasi-Monte Carlo

There is a widely believed conjecture [24] in the theory of uniform distributions that says that there exists no point sequence for arbitrary dimensionality $s$ with better star discrepancy than

$$D_n^* = O\left(\frac{(\log n)^s}{n}\right). \tag{10}$$

Sequences with this asymptotically superior discrepancy exist and are known as *quasi-random sequences* (LDSs). The points in Fig. 1(b) are drawn from one such sequence, using Sobol's construction [32]. Such sequences possess the surprising attribute that they are generated deterministically, in contrast to the pseudorandom generation of classical Monte Carlo samples. Monte Carlo performed using samples generated deterministically from a LDS is known as QMC. The overall idea is conceptually simple: rather than randomly sampling the space, we try to "fill the space" with points that are as geometrically and homogeneously equidistant as possible.

Comparing (9) and (10) gives us some sense of the possible advantages, and challenges, of the method. Comparing denominators, we see the tantalizing possibility of linear ($n^{-1}$) convergence of QMC integration error, instead of the slow ($n^{-0.5}$)

convergence of Monte Carlo error. But comparing numerators, we see that the advantages of QMC may, for larger problems (large dimensionality $s$), only make themselves apparent after a huge number of sample points $n$. Luckily, in many practical situations, this turns out not to be the case; we shall return to this in Section V.

Halton [33] provided the first method for constructing an LDS in arbitrary dimensions, by extending Hammersley's method [34] of generating finite point-sets with low-discrepancy. However, Halton's sequence suffers from very poor uniformity [24] in the high dimensions we expect for circuit applications ($10^1$–$10^2$). Sobol' [32], Faure [35], Niederreiter [36], and Niederreiter and Xing [37] have proposed sequences that were huge improvements and enabled practical use of QMC for large dimensions. These four sequences fall in a single class of LDSs called ($t$, $s$)-sequences and there constructions fall under a single general class of ($t$, $s$)-sequence constructions by Niederreiter [26], called the *digital method*. Integration lattices [38], [39] constitute another class of LDSs. Space does not permit any detailed survey of different LDS construction strategies and their theoretical properties here; please see [16] for a review. Here we use one particularly successful type of ($t$, $s$)-sequence by Sobol' for our experiments, based on the following arguments.

A comparison of the theoretical properties of the four ($t$, $s$)-sequences mentioned above suggests that the Niederreiter–Xing (NX) [37] sequences have much lower discrepancy than all others. However, there remain significant difficulties in implementing these NX sequences for arbitrary dimensions [24]. The Niederreiter sequences of [36] do not seem to offer significant improvement over the Sobol' or Faure sequences, for the general case. Given these reasons and the immense popularity of the Sobol' and Faure sequences among practitioners, we make a choice between the latter two options, for our experiments. Theoretical considerations do not provide a clear choice between these two options (see [24]). Hence, we rely on empirical observations provided in [18] and [16] which show better performance on using Sobol' sequences. Furthermore, the fact that the Sobol' sequences are in base 2 allows us to exploit fast bit-level Boolean operations in the software implementation. For these reasons, we use Sobol' sequences as our representative LDS to demonstrate the performance of QMC, and we discuss only their construction in detail.

## IV. Sobol' Points

Sobol' [32] gave the first construction of a ($t$,$s$)-sequence—he used the name $LP_\tau$-sequence. Here, we use the implementation of [40] and [41].

### A. In One Dimension

First, suppose that we are working in just one dimension ($s = 1$). We choose one primitive polynomial over the field $\mathbb{Z}_2 = \{0, 1\}$

$$x^q + d_1 x^{q-1} + \cdots + d_{q-1} x + 1 \quad d_i \in \{0, 1\} \forall i. \tag{11}$$

This is a polynomial of degree $q$ and coefficients $d_i$ in {0, 1}, satisfying two properties with respect to binary arithmetic (modulo 2):

1) it is irreducible; i.e., it cannot be factored;
2) the smallest power $p$ for which the polynomial divides $x^p + 1$ is $p = 2^q + 1$.

Tables listing primitive polynomials are widely available, for example in [42], and generation algorithms have also been suggested, as in [43]. We also choose *odd* integers $m_1, \ldots, m_q$ such that $0 < m_j < 2^j$. Now, we define *direction numbers*

$$v_j = \frac{m_j}{2^j} = 0.v_{j,1} v_{j,2} \cdots \tag{12}$$

where $v_{j,i}$ is the $i$th bit in the binary representation of $v_j$. The polynomial (11) defines a recurrence relation (in Boolean operators)

$$v_j = d_1 v_{j-1} \oplus c_2 v_{j-1} \oplus \cdots \oplus d_{q-1} v_{j-q+1} \oplus v_{j-q} \oplus \frac{v_{j-q}}{2^q} \quad j > q \tag{13}$$

where $\oplus$ denotes bitwise binary addition (modulo 2), or bitwise XOR. Note that, for a degree-$q$ polynomial, we are choosing the first $q$ direction numbers by choosing the first $q$ $m_j$ values. All other direction numbers can be computed using this recurrence relation. To compute the $n$th Sobol' value, we use

$$x_n = g_0(n) v_1 \oplus g_1(n) v_2 \oplus \cdots \tag{14}$$

where $\cdots g_2(n) g_1(n) g_0(n)$ is the *Gray code* representation of $n-1$. As suggested in [44], using the Gray code representation is much faster than using the binary representation, since only one bit changes in the Gray code form $n$ to $n+1$, making the operation (14) incremental (only one bitwise XOR), as follows. Suppose the Gray codes of $n - 1$ ($\{g_i(n)\}$) and $n$ ($\{g_i(n+1)\}$) differ in the $l$th bit. Then, we can write

$$x_{n+1} = x_n \oplus v_l. \tag{15}$$

### B. Multiple Dimensions: Choosing Primitive Polynomials and Direction Numbers

For a general problem with $s > 1$ dimensions, we choose $s$ *different* primitive polynomials, one for each coordinate. The method above is used to generate sequences for each coordinate. For best discrepancy properties, Sobol' [32] suggests choosing primitive polynomials with the smallest possible degree. Hence, for increasing dimension, unique polynomials are chosen sequentially with nondecreasing smallest possible degree $q$.

One additional problem is how to choose the initial $m_j$ values in (12) for each dimension $i$. Sobol' shows that the uniformity properties of the sequence also depend on the choice of the direction numbers $v_j$, and, hence, the $m_j$ values. Let us denote the $j$th direction number for the $i$th dimension by $v_j^{(i)}$. Let $v_{j,1}^{(i)}$ denote the first bit of $v_j^{(i)}$. Set the matrix

$$V_s = [v_{j,1}^{(i)}] \text{ where } 1 \le i \le s \ 1 \le j \le s. \tag{16}$$

Then, according to Sobol's development in [32], the condition

$$|V_s| = 1 (\text{mod } 2) \tag{17}$$

gives better uniformity. This is Sobol's "property A." Hence, $m_j^{(i)}$ are chosen to satisfy this property (see [41]).

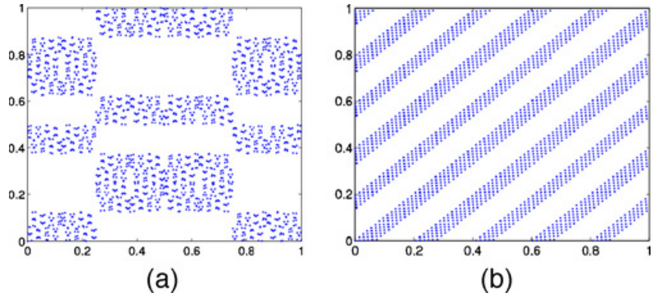

Fig. 2. Undesirable patterns in 2-D projections of high-dimensional LDSs. (a) Dimensions 38 and 39 of a 40-D Sobol' sequence: first $2^{11}$ points. (b) Dimensions 1 and 30 of a 70-D Faure sequence: first $2^{11}$ points.

A generator for Sobol' points is relatively straightforward to implement, requiring mainly bit-level Boolean operations, and relatively little of the number-theoretic difficulty of some of the other LDS strategies [37]. However, all LDS schemes suffer from some idiosyncrasies when applied to higher dimensional problems, requiring additional care in the way we map out statistical integration problem into a viable QMC formulation. We treat these issues in the following section.

## V. QMC in High Dimensions

Looking only at the asymptotics of the discrepancy (10), the $O((\log n)^s/n)$ error bound of QMC should not show any improvement over the $O(n^{-0.5})$ bound for conventional Monte Carlo, in the case of large $s$ and only feasibly large sample size $n$. It would take very large sample sizes before the $O(n^{-1})$ error convergence can manifest. Mechanistically, any LDS requires increasingly large number of points to uniformly fill the unit cube as the dimensions increase. Suppose we generate a $s = 40$ dimensional Sobol' point-set, pick any two coordinates and plot the values of those coordinates. If the points are uniform in 40 dimensions, we expect to see a 2-D projection like Fig. 1(b). However, we often observe high-discrepancy 2-D projections, as shown in Fig. 2(a). These undesirable patterns are seen for all LDSs. Fig. 2(b) shows the case of a 2-D projection from a 70-D Faure sequence. These patterns always fill out uniformly as more points are sampled from the sequence, but this often requires impractically large number of points. The theoretical reasons for why these patterns occur are well known, but are skipped here for lack of space (see [24]).

In spite of this degraded uniformity for large $s$, QMC has been seen to outperform Monte Carlo even for problems with very large $s$, e.g., IBM's 1439-dimensional derivative pricing experiments of [19]. This anomalous, empirical success has been largely explained using the concept of *effective dimension* [30]. We review this concept here because it strongly impacts the manner in which we, or, we believe, most practitioners, will map circuit problems into a successful QMC form.

### A. ANOVA Decomposition and Effective Dimension

The effective dimension concept captures the dependence of our integrand $f(\mathbf{x})$ on subsets of variables in $\mathbf{x} = \{x_1, \ldots, x_s\}$. To study these properties, we use the tool of the *analysis of variance* (ANOVA) decomposition of $f$, that represents $f$ as a combination of simpler functions defined over variable subsets. Consequently, it lets us separate out the various contributors to the variation in $f$, in terms of smaller dimensional subspaces. We will exploit this capability to separate out the 1-D and multidimensional parts of the integrands for various circuit analysis problems, and to understand why QMC performs so well on these problems in comparison with Monte Carlo and LHS.

Assume in (1) that function $f \in L_2(C^s)$ is square integrable over $C^s$. Let $\mathbf{u} \subseteq \{1, \ldots, s\}$ denote any subset of the input dimensions of $f$. We use $|\mathbf{u}|$ for its cardinality and $-\mathbf{u}$ for its complementary set $\{1, \ldots, s\} - \mathbf{u}$. Then, for any point $\mathbf{x} = \{x_1, \ldots, x_s\} \in C^s$, $\mathbf{x}_{\mathbf{u}} = \{x_i : i \in \mathbf{u}\}$ is the vector of the coordinates of $\mathbf{x}$ belonging to $\mathbf{u}$. $C^{\mathbf{u}}$ is the unit cube in the dimensions belonging to $\mathbf{u}$. We can write $f$ as the sum of $2^s$ "simpler" functions using its ANOVA decomposition as

$$f(\mathbf{x}) = \sum_{\mathbf{u}\subseteq\{1,\ldots,s\}} f_{\mathbf{u}}(\mathbf{x}) \tag{18}$$

where each function $f_{\mathbf{u}}$ depends only on $\mathbf{x}_{\mathbf{u}}$, excluding the effect of the proper subsets of $\mathbf{x}_{\mathbf{u}}$. The ANOVA terms are defined using the recurrence relation

$$f_{\mathbf{u}}(\mathbf{x}) = \int_{C^{-\mathbf{u}}} f(\mathbf{x}) d\mathbf{x}_{-\mathbf{u}} - \sum_{\mathbf{v}\subset\mathbf{u}} f_{\mathbf{v}}(\mathbf{x}). \tag{19}$$

This relation performs a simple task: remove all parts of $f$ that are not functions of exactly the variables in the subset $\mathbf{u}$. The first term integrates over all variables not in the subset $\mathbf{u}$, and is hence, a function only of variables in $\mathbf{u}$. The second term is a sum of functions that depend on only proper subsets of $\mathbf{u}$. The difference between the first and second terms, then gives that part of $f$ that is a function of exactly the variables in $\mathbf{u}$.

The ANOVA decomposition is orthogonal: $\int_{C^s} f_{\mathbf{u}} f_{\mathbf{v}} d\mathbf{x} = 0$ if $\mathbf{u} \neq \mathbf{v}$. Hence, the variance of $f$ [defined in (4)] can be written as

$$\sigma^2 = \sum_{|\mathbf{u}|>0} \sigma_{\mathbf{u}}^2 \text{ where } \sigma_{\mathbf{u}}^2 = \int_{C^s} f_{\mathbf{u}}(\mathbf{x})^2 d\mathbf{x} \tag{20}$$

is the variance of $f_{\mathbf{u}}$. Now, taking $p$ as a selected probability value close to 1 (typically 0.99), we have the following definitions of effective dimension [30].

**Superposition sense**: *The effective dimension of $f$ with variance $\sigma^2$, in the superposition sense, is the smallest integer $s_S$ such that*

$$\sum_{0<|\mathbf{u}|\leq s_S} \sigma_{\mathbf{u}}^2 \geq p\sigma^2. \tag{21}$$

**Truncation sense**: *The effective dimension of $f$ with variance $\sigma^2$, in the truncation sense, is the smallest integer $s_T$ such that*

$$\sum_{\mathbf{u}\subseteq\{1,\ldots,s_T\}} \sigma_{\mathbf{u}}^2 \geq p\sigma^2. \tag{22}$$

In other words, $s_T$ is the number of leading dimensions, in a fixed ordering, that account for most of the variance in the function value, while $s_S$ is an indicator of whether only low-dimensional interactions dominate the variation in $f$. For example, $f(\mathbf{x}) = x_1 + x_2 + x_4$ has truncation dimension $s_T = 4$, but superposition dimension $s_S = 1$.

### B. Variable-Dimension Mapping for Effective QMC

Effective dimension is relevant to us because it has been largely invoked to reason why QMC has been so efficient (e.g., 150× speedup [19]) on large financial problems, in spite of degraded high-dimensional discrepancy of the LDS. Extensive experiments showing the advantage of exploiting reduced effective dimension can be found in [18], [30], [45], and [46]. We will exploit this same reasoning to effectively map circuit problems into QMC form. To start, we pose the question, how are quadrature point-set discrepancy and effective dimension of the integrand related to integration error?

To answer this question, we return to the Koksma–Hlawka bound (7), but use an ANOVA version of it

$$|Q - Q_n| \leq \sum_{\mathbf{u} \subseteq \{1,\ldots,s\}:|\mathbf{u}|>0} V_{\mathbf{u}}(f_{\mathbf{u}}) D^*_{n,\mathbf{u}} \tag{23}$$

where $V_{\mathbf{u}}(f_{\mathbf{u}})$ is the variation of $f_{\mathbf{u}}$ taken as a $|\mathbf{u}|$-dimensional function over $C^{\mathbf{u}}$, and $D^*_{n,\mathbf{u}}$ is the star discrepancy of the $|\mathbf{u}|$-dimensional points obtained by projecting the point-set onto the coordinates in $\mathbf{u}$. For a derivation, see [24]. From this relation, if, for the subsets $\mathbf{u}$ that cause large variance $\sigma_{\mathbf{u}}$ in $f$ [captured by $V_{\mathbf{u}}(f_{\mathbf{u}})$], the discrepancy $D^*_{n,\mathbf{u}}$ of the projection of the LDS sample onto $\mathbf{u}$ is small, then all the terms in the error bound are small, leading to a small error bound. This suggests a notion of a good *variable-dimension mapping*: map the important subset of variables [large $\sigma_{\mathbf{u}}$ or $V_{\mathbf{u}}(f_{\mathbf{u}})$] to the LDS dimensions with good uniformity (low $D^*_{n,\mathbf{u}}$). Of course, this is viable only if these important subsets are not large, since for large dimensions (large $|\mathbf{u}|$) $D^*_{n,\mathbf{u}}$ will also be large. Fortunately, as suggested in other research [11], [14] and as we shall see in this paper, circuit problems tend to have low latent dimensionality. Consequently, these important subsets are often of small size.

There can be two mechanisms for exploiting any low effective dimension properties of the integrand in our circuit analysis problems. First, if the truncation dimension $s_T$ of $f$ is small, the error bound (23) for LDS can be reduced by using an LDS that is uniform in the first $s_T$ dimensions, even if the higher dimensions are not sampled uniformly. Typically, the initial dimensions are sampled more uniformly by an LDS than by pseudorandom points. Hence, $D^*_{n,\mathbf{u}}$ can be small for $\mathbf{u} = \{1, \ldots, s_T\}$ if $s_T$ is not too large. For example, the initial dimensions (e.g., dimensions 1–10) of a typical Sobol' sequence have lower discrepancy than the later dimensions (e.g., dimensions 91–100), because lower degree primitive polynomials typically yield more uniform sampling [24], [32]. Second, if the superposition dimension $s_S$ of $f$ is also small, the error bound (23) may be smaller than pseudorandom points. On average, the low dimension projections of LDSs can show better uniformity (smaller $D^*_{n,\mathbf{u}}$) than pseudorandom points, even for realistic sample sizes [46], particularly in the earlier dimensions. Hence, this mechanism is useful if $s_T$ is not large (e.g., $s_T \leq 10$).

We aim to use the first mechanism, but how do we ensure a small $s_T$ for our circuit problems? For problems with a time-series random-walk structure, there are good techniques for reducing $s_T$ [46], but these are not applicable in the case of circuit yield analysis. Principal component analysis (PCA) may be useful here to sort the variables by decreasing variance contribution, but does not exploit any measure of variable influence on $f$. Here we note that recent research results [11], [14] have suggested that circuit performance metrics are largely sensitive to only a few variables. If we sort the variables in decreasing order of this "sensitivity" or "importance," only the first few variable can capture all the variation in the integrand. In other words, this sorting will give us small truncation dimension $s_T$. We can then directly use an LDS on this sorted set of variables. Note that we are essentially mapping important variables to the more "uniform" dimensions of the LDS.

We can suggest two sorting strategies.

1) The designer selects the parameters that most affect the relevant performance metrics, and assigns these to the lower coordinates of the QMC. This can be a feasible option in manual design settings where the statistical parameters correspond to different devices in the circuit, since circuit designers often have good insight regarding the devices that significantly affect the relevant performance metrics.
2) The global sensitivity of the metric to circuit parameters is used a measure of their importance, and the parameters are sorted in decreasing order of importance. This sorted list is then mapped to the corresponding LDS coordinates.

We use the latter method here. The measure of sensitivity we use is the absolute value of Spearman's rank correlation coefficient [47]. This is similar to Pearson's linear correlation, but is more robust in the presence of nonlinearity. Suppose $R_j$ and $S_j$ are the ranks of corresponding values of a variable and a circuit performance metric, then the rank correlation is given as

$$\rho_S = \frac{\sum_{j=1}^{n}(R_j - \bar{R})(S_j - \bar{S})}{\sqrt{\sum_{j=1}^{n}(R_j - \bar{R})^2}\sqrt{\sum_{j=1}^{n}(S_j - \bar{S})^2}}. \tag{24}$$

This rank correlation is computed by first running a small Monte Carlo run. For multiple metrics, we use the sum of rank correlation values across all metrics as the measure of importance for any variable.

## VI. Latin Hypercube Sampling

LHS, introduced in [48], is a popular variance reduction technique based on stratification applied to Monte Carlo. Recalling the asymptotic Monte Carlo variance in (5), LHS reduces this variance by reducing the contribution of $\sigma(f)$.

### A. Construction

Suppose we wish to generate an $n$-point LHS sample over $C^s$. Define $\{\pi_j\}_{j=1}^s$ as $s$ independent and random permutations of $\{1, \ldots, n\}$. Also, let $u_{ij}$ ($i = 1, \ldots, n$; $j = 1, \ldots, s$) be $n \times s$ i.i.d. random variables distributed uniformly over [0, 1). Then, the $j$th coordinate of the $i$-point in the $n$-point LHS sample is

$$x_{ij} = \frac{\pi_j(i) - 1 + u_{ij}}{n} \quad i = 1, \ldots, n \;\; j = 1, \ldots, s. \tag{25}$$

In effect, we create $n$ equal slices or strata of [0, 1) along each dimension, select a random value within each stratum for every coordinate, and then randomly match up the coordinate values to generate $n$ $s$-dimensional points. In this manner, LHS ensures high uniformity of the point-set along any one dimension.

### B. Why LHS is Better than Monte Carlo

The stratification in LHS ensures that the sample points are always well spread out over the unit cube. Compared to Monte Carlo then, there is less variation in the way the integrand $f$ is sampled by different LHS samples of same size. As a result, we expect less variation in the integral estimate $Q_n$ compared to conventional Monte Carlo. We can express this more formally using ANOVA decomposition. Using the orthogonality of ANOVA decomposition, we can write the overall function variance as

$$\sigma^2 = \sigma_1^2 + \sigma_{>1}^2 \;:\; \sigma_1^2 = \sum_{|\mathbf{u}|=1} \sigma_{\mathbf{u}}^2,\; \sigma_{>1}^2 = \sum_{|\mathbf{u}|>1} \sigma_{\mathbf{u}}^2 \tag{26}$$

where $\sigma_1^2$ is the variance of the 1-D part of $f$ given by $g_1(\mathbf{x}) = \sum_{|\mathbf{u}|=1} f_{\mathbf{u}}(\mathbf{x})$ and $\sigma_{>1}^2$ is the remaining variance of $f - g_1$. From [49], we know that the asymptotic variance of the LHS estimate of $Q$ is

$$\sigma_{\text{LHS}}^2 = \frac{\sigma_{>1}^2}{n} + o\left(\frac{1}{n}\right). \tag{27}$$

Hence, compared to the Monte Carlo estimate (5), LHS achieves a variance reduction by reducing the variance in estimating the integral of the 1-D part of $f$ to $o(n^{-1})$. On the remaining ($>$ 1-D) part of $f$, LHS shows Monte Carlo performance. An LHS sample can be shown to be essentially a type of "scrambled" low-discrepancy point-set. A derivation of this claim is beyond the scope of this paper (see [24]), but it is relevant because under certain smoothness criterion, the second term on the right side of (27) reduces as $O(n^{-3})$ [50] for such scrambled LDS point-sets. Indeed, if $f$ is largely 1-D; i.e., it has an effective dimension ($s_S$ or $s_T$) of one, LHS can be an excellent quadrature technique. We then must ask, is QMC better than LHS?

### C. Why QMC is Better than LHS

Let $f$ have a truncation dimension $s_T$ of 24: variables $x_1$ and $x_2$ have the maximum impact on $f$. We know that LHS can largely remove the variance of $f$ due to the 1-D components, resulting in tighter estimation error. However, it still shows Monte Carlo behavior on the part of $f$ that depends on both $x_1$ and $x_2$. We anticipate, from the low discrepancy of a Sobol' sequence in the first two dimensions, that the error may be further reduced by using Sobol' points.

Numerical experiments on synthetic functions from [51] indicate why QMC, and Sobol' points in specific, can outperform LHS. Combining our arguments above with these numerical results, we can enumerate two types of functions for which Sobol' points may provide improved integration errors compared to LHS, along with the relevant features of Sobol' points.

1) *Most variance contributed by 1-D ANOVA components.* This is the class of functions for which LHS provides large improvements over standard Monte Carlo. Numerical results in [51] show that the discrepancies of 1-D projections of Sobol' points are even better than for LHS. This allows us to retain the advantages that LHS provides: low variation in integration of the 1-D parts of the integrand $f$.
2) *Significant contribution from higher dimensional ANOVA components with small truncation dimension $s_T > 1$.* Now, we allow higher dimensional ANOVA components to have a large contribution to the function variance, as long as the corresponding dimensions are from the early dimensions of the point set. Clearly, LHS provides no extra advantage beyond that for the 1-D projections. Sobol' points, however, do. The discrepancy of the projection of Sobol' points onto some subset $\mathbf{u}$, with small $|\mathbf{u}| > 1$, tends to be lower than that for LHS, as long as the subset is from the early dimensions; i.e., $\mathbf{u} \subset \{1, \ldots, l\}$ for $l \lesssim 10$. These conditions on $f$ are significantly less restrictive in practice than those for LHS being the best option, and suggest that Sobol' sequences—and any other competitive LDS—can perform better than, or as well as, LHS in general.

[24] and [51] provide further theoretical and experimental evidence for these arguments. In this paper (Section VIII), we directly test them on our circuit benchmarks and, indeed, observe conforming results.

### D. Latent Dimensionality and LHS

Combining (5) and (27), for large sample size $n$, we can write

$$\frac{\sigma_1^2}{\sigma^2} = 1 - \frac{\sigma_{>1}^2}{\sigma^2} \approx 1 - \frac{\sigma_{\text{LHS}}^2}{\sigma_{\text{MC}}^2}. \tag{28}$$

If we estimate $\sigma_{\text{LHS}}$ and $\sigma_{\text{MC}}$ by taking sample variance across estimates from several LHS and Monte Carlo runs, respectively, we can then estimate the contribution of 1-D ANOVA terms to the variance of $f$. In other words, we can estimate how much of $f$ is 1-D. Such information provides insight into the structure of circuit problems, which may then be exploited for further research on circuit modeling and analysis [11], [14].

This estimate also gives us a way to explain observed relative performances of Monte Carlo, LHS and QMC on any test case. We will use it in Section VIII to analyze the results of our experiments, in the context of our preceding discussion comparing LHS with Monte Carlo and QMC.

## VII. Estimating Integration Error

In practice, the exact value of $Q = \int f(\mathbf{x})d\mathbf{x}$ is unknown, for example the exact value of circuit yield. Then how do we estimate the error in the quadrature estimate $Q_n$? Random methods (Monte Carlo and LHS) make this easy since the sample standard deviation of the estimate across several runs can be used as a probabilistic measure of the error. QMC, however, is a deterministic technique: we get the same estimate $Q_n$ every time we run an $n$-point QMC, assuming no change in the initialization parameters of the LDS (e.g., primitive polynomials for Sobol' points). There is no natural variance that we can exploit to estimate the error. Also, bounds on the error, like the Koksma–Hlawka bound (7), are unusable because of at least two reasons: 1) it is usually computationally infeasible to estimate both $V(f)$ and $D_n^*$ with acceptable accuracy, and 2) the error *bound* can be very different from the actual error value. One way to solve this problem is to artificially randomize the QMC points while maintaining two important properties.

1) Every point in the scrambled set has a uniform distribution over $C^s$, so that the approximation $Q_n$ is unbiased.
2) The resulting point-sets still possess the same theoretical uniformity properties [26] as the original LDS point-set.

We can then use several different randomized QMC (RQMC) runs to compute a sample standard deviation of the integration estimate, just like for Monte Carlo.

### A. Scrambled LDS for Randomized QMC

Owen [52] proposes a general *scrambling* scheme for randomizing $(t,\ s)$-sequences. Let $\{\mathbf{x}_1, \mathbf{x}_2, \dots\}$ and $\{\mathbf{z}_1, \mathbf{z}_2, \dots\}$ denote the original sequence and a randomly scrambled version, respectively, both in base $b$ ($b = 2$ for Sobol' points). Let $x_n^{(i)} = 0.x_{n,1}^{(i)}x_{n,2}^{(i)}\cdots$ be the $i$th coordinate value of $\mathbf{x}_n$: $x_{n,j}^{(i)}$ is the $j$th digit (bit for Sobol' points) of this coordinate value. Assuming similar definitions for $z_n^{(i)}$ and $z_{n,j}^{(i)}$, the scrambling relations are

$$z_{n,1}^{(i)} = \pi^i(x_{n,1}^{(i)}) \;\; z_{n,j}^{(i)} = \pi^i_{x_{n,1}^{(i)},x_{n,2}^{(i)},\dots,x_{n,j-1}^{(i)}}(x_{n,j}^{(i)}) \tag{29}$$

where $\pi^i_{\dots}$ are random permutations of the digits $\{0, 1, \dots, b-1\}$ in base $b$ ($\{0, 1\}$ for Sobol' points), chosen uniformly and mutually independently. The permutation chosen for the $j$th digit $x_{n,j}^{(i)}$ depends on the precise values of the previous $j-1$ digits; this is denoted symbolically with the $x_{n,1}^{(i)}, x_{n,2}^{(i)}, \dots, x_{n,j-1}^{(i)}$ subscript in (29). Other schemes have also been proposed [53].

Owen's original scrambling uses a large amount of memory. Hence, we use a less powerful, but more scalable, *linear matrix scrambling* version from [54]. Both methods preserve the two desired properties from above. The main difference of the latter method from Owen's method is that the number of scramblings possible is reduced, although it is still much larger than what is needed for most practical purposes. [54] presents a general version, but we present it only in the terminology of the Sobol' construction of Section IV. The idea consists of two simple steps.

1) Replace the original direction vectors $\mathbf{v}_j$ by new scrambled direction vectors $\mathbf{v}'_j$, given as

$$\mathbf{v}'^{(i)}_j = v_{j,1}^{(i)} \cdot \mathbf{l}_1^{(i)} \oplus v_{j,2}^{(i)} \cdot \mathbf{l}_2^{(i)} \oplus \cdots \quad 1 \le i \le s \;\; j > 0 \tag{30}$$

where $\mathbf{l}_k^{(i)}$ are vectors in base 2 such that the matrices $\mathbf{L}^{(i)} = [\mathbf{l}_1^{(i)} \;\; \mathbf{l}_2^{(i)} \;\; \cdots]$ are chosen to be random, mutually independent, nonsingular, and lower triangular.

2) Add (modulo 2) randomly and independently chosen base-2 vectors $\mathbf{e}^{(i)}$ to each Sobol' point

$$z_n^{(i)} = \mathbf{e}^{(i)} \oplus g_0(n)v'^{(i)}_1 \oplus g_1(n)v'^{(i)}_2 \oplus \cdots \tag{31}$$

where $\cdots g_2(n)g_1(n)g_0(n)$ is the Gray code representation of $n-1$. In fact, we need to XOR $\mathbf{e}^{(i)}$ only once, to the first point, and subsequent points are given simply as

$$z_{n+1}^{(i)} = z_n^{(i)} \oplus v'^{(i)}_l \tag{32}$$

where $l$ is the bit index where the Gray codes of $n$ and $n-1$ differ.

Now we are well equipped to demonstrate the performance of QMC using experiments and to compare it with Monte Carlo and LHS.

## VIII. Experimental Results

In this section we compare the performance of QMC with conventional Monte Carlo and LHS on three different test cases. First, though, we briefly mention some relevant implementation details.

1) [18] Suggests that LDSs perform better when the first few points are skipped. As recommended there, we skip the first $n_{\text{skip}} = 2^{\lfloor \log_2 n \rfloor}$ Sobol' points.
2) A linear congruential generator (LCG) [16] [`drand48()` in C] is used to generate the pseudorandom sequences for standard Monte Carlo and LHS, and for random scramblings of the Sobol' points. This generator enjoys widespread popularity and the obtained results will be immediately relevant to the general practitioner. Also, variance results in [55] comparing LCG with a generalized feedback shift register generator (GFSR) [56], do not show significant improvements for GFSR in the context of randomized QMC.
3) The standard Box Muller [57] method for generating normally distributed variates can be inaccurate, especially for a large number of samples [15]. Hence, an inverse transform method, by Acklam [58], is used.
4) For each test case ten Monte Carlo runs, one QMC run with standard Sobol' points and nine RQMC runs with scrambled Sobol' points are run. LHS sample generation is not incremental, so ten LHS runs with four different sample sizes are run.

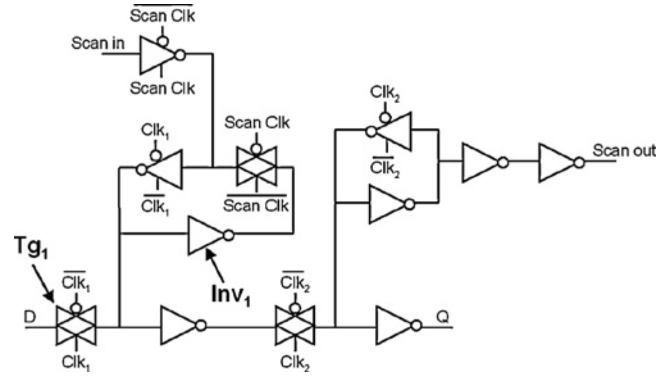
$\overline{\text{Scan Clk}}$
Scan in
Scan Clk
$\text{Clk}_1$
Scan Clk
$\overline{\text{Clk}_1}$
$\overline{\text{Scan Clk}}$
$\text{Clk}_2$
$\overline{\text{Clk}_2}$
Scan out
$\text{Tg}_1$
$\overline{\text{Clk}_1}$
$\text{Inv}_1$
$\overline{\text{Clk}_2}$
D
Q
$\text{Clk}_1$
$\text{Clk}_2$

Fig. 3. Master-slave flip–flop with the scan chain component.

Now we describe the benchmark circuits and the experiments. All samples were evaluated using detailed circuit simulation in Cadence Spectre. Results for all test cases will be analyzed together in Section VIII-D.

### A. Master-Slave Flip–Flop with Scan Chain (MSFF)

The first test case is a commonly seen master-slave flip–flop with scan chain shown in Fig. 3, which we refer to as simply MSFF. The circuit has been implemented using the 45 nm CMOS Predictive Technology Models of [59]. The variations considered are random dopant fluctuation (RDF) for all transistors and one global gate oxide thickness ($t_{ox}$) variation. The RDF is modeled as normally distributed independent threshold voltage ($V_t$) variation

$$\delta V_t \sim \mathcal{N}\left(0, \left(\frac{13.5 V_{t0}}{\sqrt{WL}}\right)^2\right) \tag{33}$$

where $W, L$ are the transistor width and length in nanometer, and $V_{t0}$ is the nominal threshold voltage. This results in about 30% standard deviation for a minimum-sized transistor. The standard deviation for $t_{ox}$ is taken as only 2% of the nominal value, since $t_{ox}$ is typically much better controlled than RDF. The number of statistical parameters, $s$, is 31. The metric we measure is the clock-output delay, $\tau_{cq}$ and estimate the integral of the parametric yield, given a maximum acceptable delay of $\tau_{\max} = 200$ ps. To see how parametric yield fits the canonical form of (1), we define the pass–fail indicator function for a threshold $t$ as

$$I_t(\mathbf{z}, g(\mathbf{z})) = \begin{cases} 1 \text{ (pass)} & g(\mathbf{z}) \leq t \\ 0 \text{ (fail)} & g(\mathbf{z}) > t \end{cases} \tag{34}$$

where $g$ is some performance metric (e.g., $\tau_{cq}$), and write parametric yield as

$$Y_t(g, \Pi) = \int_{C^s} I_t(\Pi^{-1}(\mathbf{x}), g(\Pi^{-1}(\mathbf{x})))d\mathbf{x} \tag{35}$$

where $\Pi^{-1}(\mathbf{x})$ transforms uniformly distributed $\mathbf{x} \in C^s$ to the required distribution of $\mathbf{z}$ (normal in this case). For our MSFF example, the yield is given as $Y_{\tau_{\max}}(\tau_{cq}, \Phi)$. The Monte Carlo and QMC sample sizes are 50 000 and the LHS sample sizes are 312, 1250, 5000, and 50 000.

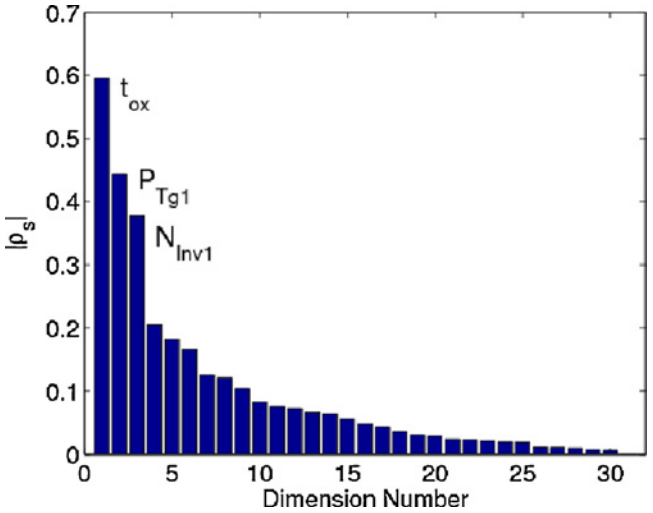

Fig. 4. MSFF parameters sorted by decreasing importance (magnitude of rank correlation with the flip–flop delay).

The results are discussed in Section VIII-D, but as an illustrating example, here we look at how the rank correlation based variable-dimension mapping works for this test case. Fig. 4 shows the magnitude of the rank correlation ($|\rho_S|$) of each parameter with the clock-output delay for rising output, computed from an initial Monte Carlo run of 1000 samples. The variables are sorted in *decreasing* order of importance (rank correlation magnitude): this is the order in which they will be mapped to the increasing dimensions of the Sobol' sequence. The three most important parameters are labeled: 1) $t_{ox}$: global gate oxide variation; 2) $P_{Tg1}$: the $V_t$ variation in the pMOS device in the input transmission gate $\text{Tg}_1$; and 3) $N_{Inv1}$: the $V_t$ variation in the nMOS device in the inverter $\text{Inv}_1$. The latter two devices are on the critical signal path for a high input causing a rising output, and are important for correctly sampling a "1" at the input, especially when the input timing is close to the setup limit. Since the input was timed in such a manner in the testbench, these measures of importance make intuitive sense.

### B. 64-Bit SRAM Column

Yield analysis of SRAMs is unavoidable today, given the large SRAM capacity and large variations due to random dopant fluctuation. Our next test case is a 64-bit SRAM column, with nonrestoring write driver and column multiplexer (Fig. 5). Only one cell is being accessed, while all the other wordlines are turned off. Random threshold variation on all 402 transistors (including the write driver and column mux) are considered, along with a global gate oxide variation, giving us a total of $s = 403$ variables. All variations are assumed to be normally distributed and the $V_t$ standard deviation is taken as

$$\delta V_t \sim \mathcal{N}\left(0, \left(\frac{5mV}{\sqrt{WL}}\right)^2\right) \tag{36}$$

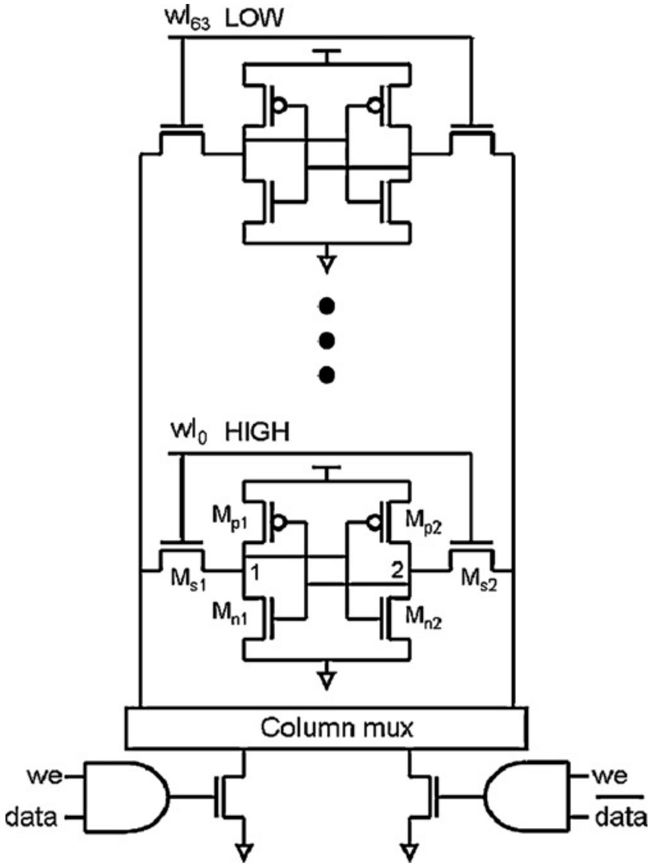

Fig. 5. 64-Bit SRAM column with write driver and column multiplexer.

where $W, L$ are the transistor width and length in $\mu$m. The standard deviation for $t_{ox}$ is taken as 2% of the nominal value.

The metric measured here is the write time $\tau_w$: the time between the wordline going high, to the nondriven cell node (node 2) transitioning. Here "going high" and "transitioning" imply crossing 50% of the full voltage swing. The write time is computed as a multiple of the fanout-4 delay of an inverter (FO4). The quantity being estimated is the 90th percentile of the write time. Although the percentile is estimated using a search on the sample values, we can theoretically write it as a closed form integral. If we define the following step function:

$$H_p(\mathbf{x}, g, \Pi) = \begin{cases} 0 & Y_{g(\Pi^{-1}(\mathbf{x}))}(g, \Pi) < \frac{p}{100} \\ 1 & \text{otherwise} \end{cases} \tag{37}$$

then we can write any $p$th percentile as

$$g_p(g, \Pi) = \int_{C^s} g(\Pi^{-1}(\mathbf{x})) dH_p(\mathbf{x}, g, \Pi) \tag{38}$$

which is in the Stieltjes integral version of the canonical Riemann form (1). This expression cannot be written in the Riemann form because the derivative of $H_p$ is unbounded [60]. However, it is clear that the percentile computation can be written as an integral computation problem. In our case we are estimating $g_{90}(\tau_w, \Phi)$. The Monte Carlo and QMC sample sizes are 10 000 and the LHS sample sizes are 156, 625, 2500, and 10 000. Results are presented in Section VIII-D.

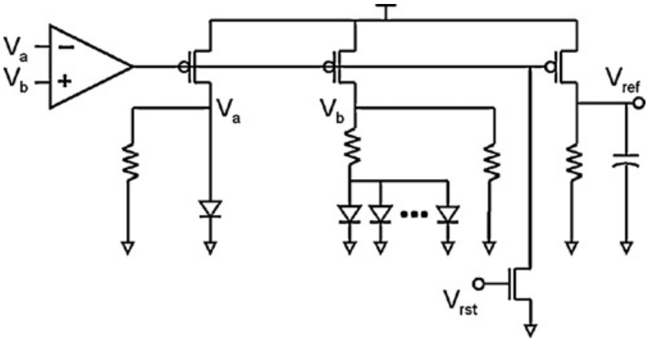

Fig. 6. Low-voltage CMOS bandgap voltage reference circuit from [61], with a parameter space of 122 dimensions.

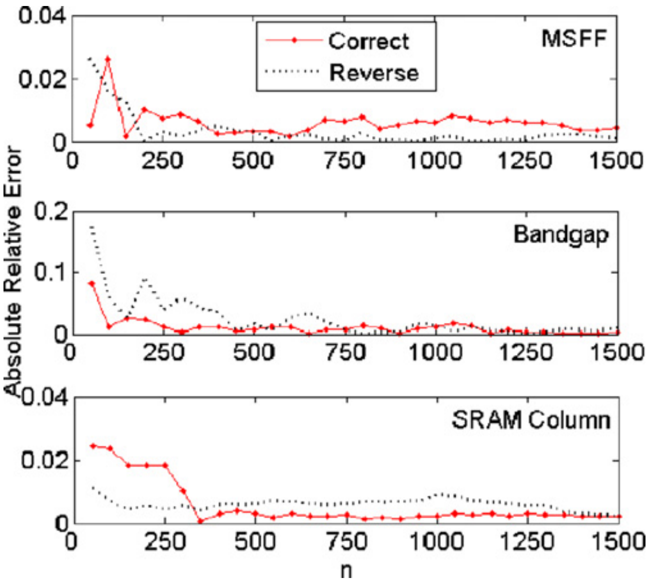

Fig. 7. QMC estimate with increasing number of points, for the correct variable-dimension mapping (variables sorted with decreasing rank correlation: solid with dots), and for the reverse mapping (increasing rank correlation: dotted).

## C. Sub-1-V CMOS Bandgap Voltage Reference

Fig. 6 shows a low-voltage CMOS bandgap voltage reference circuit, proposed in [61]. The circuit is able to provide reference voltages that are less than 1-V, and is built using standard CMOS technology. It was chosen for its relevance in today's and tomorrow's low-voltage designs, and also because the related RSM problem has a high input dimensionality of 122 and strong nonlinear behavior. The op amp used is a standard single-ended RC-compensated two-stage op amp [62]. The circuit has 101 diodes. The transistor device and variation models are the same 90 nm CMOS as the SRAM. Variations in each diode are modeled as a normally distributed variation on the saturation current, with standard deviation of 10%. Each resistor and capacitor has its own normally distributed variation source, with a standard deviation of 5%. There are a total of 121 local variation parameters and one global $t_{ox}$ variation.

In this case, we measure three metrics: 1) output voltage ($V_{\text{ref}}$); 2) settling time ($\tau_s$); and 3) dropout voltage ($V_{\text{do}}$) $V_{\text{do}}$ is the difference between the supply voltage and $V_{\text{ref}}$ when

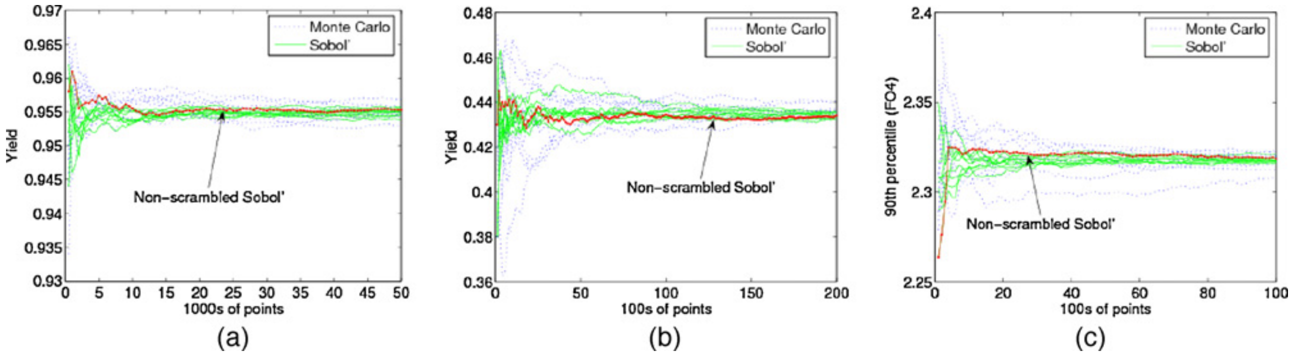

Fig. 8. Estimates from Monte Carlo (pseudorandom: dotted) and QMC (scrambled Sobol': solid, nonscrambled Sobol': bold). (a) MSFF yield estimates (exact value $\simeq$ 0.9549). (b) Bandgap yield estimates (exact value $\simeq$ 0.4353). (c) SRAM 90th percentile estimates (exact value $\simeq$ 2.3176).

the latter falls by 1% of its nominal value (0.6-V): lower $V_{\text{do}}$ implies a more robust circuit. The circuit performance is deemed acceptable only if $V_{\text{ref}}$ is within 10% of 0.6-V, $\tau_s \leq$ 200 ns and $V_{\text{do}} \leq$ 0.9-V. The Monte Carlo and QMC sample sizes are 20 000 and the LHS sample sizes are 312, 1250, 5000, and 20 000.

### D. Analysis of Results

First, we look at the Monte Carlo and QMC runs. Fig. 8 plots the values of the integral estimates with increasing number of points for each Monte Carlo (pseudorandom) and QMC (Sobol') run. The captions show our best guess of the exact value $Q$ as estimated using all points at our disposal from the ten Monte Carlo runs, ten QMC runs, and the ten largest LHS runs. For all three cases, we can clearly see that the QMC graphs converge more quickly than the Monte Carlo graphs in general. In particular, the *nonscrambled* Sobol' points converge very fast toward the final result. This fact provides indirect validation that our rank correlation based dimension mapping is an effective heuristic. Scrambling the digits of an LDS sample changes the way the sampling space is filled up, and hence, changes the patterns and the discrepancies of the projections of the point set. We observe here that changing the patterns in this way causes the QMC performance to degrade in general for our benchmarks. This implies that the rank correlation arranges the variables in a way that is optimal (or at least, advantageous), given the patterns of the nonscrambled LDS. In [11], we conjectured that since this behavior gets more pronounced from MSFF to the bandgap to the SRAM Column, it may be that for low dimensionality (e.g., 31-dimensional MSFF), the LDS uniformity does not show large variation for different projects, while for high dimensionality, effective variable-dimension mapping may give more notable improvement. Although this argument may be true, we will shortly see another, more convincing explanation emerge from the results in this paper, specifically when we analyze in the context of the LHS convergence results.

Fig. 7 plots the absolute values of the relative QMC estimate errors with two different variable orderings: 1) correct—we use the rank correlation method, and 2) reverse—we use the reverse ordering with increasing rank correlation as the dimensions increase. The motivation for this experiment is to compare our heuristic for a "best case" ordering with a guess for the "worst case" ordering. Note that the reverse ordering is not necessarily the worst, because the degradation of low-dimensional projection discrepancy is not monotonically increasing with dimension. The results concur with what we see in Fig. 8: the advantage of the "correct" ordering becomes more apparent as we move from the MSFF to the bandgap to the SRAM Column. We will shortly see an explanation for this observed behavior.

1) *Convergence Comparison:* Fig. 9 compares the standard deviation of the Monte Carlo, LHS and QMC estimates with increasing number of points. The plots are in $\log_{10}-\log_{10}$ scale, where a $\sigma \propto n^{-\alpha}$ relationship will appear as a straight line with slope $-\alpha$. Linear fits, via least squared error, are shown as dashed straight lines, and are annotated with the corresponding convergence exponent. Let us compare QMC and Monte Carlo first. We can immediately see that QMC shows lower variance and faster convergence than Monte Carlo across all three benchmarks. The estimated Monte Carlo convergence rates are a little slower than the asymptotic rate of $n^{-0.5}$. This can be because we have not reached the asymptotic rate. Even if the convergence rate was to reach this theoretical limit for large $n$, QMC will still exhibit faster convergence for all test cases ($\alpha_{\text{QMC}}$ >0.5). Furthermore, with increasing $n$, $\alpha$ for QMC should only tend further toward 1.0. The observed convergence rates between the asymptotes of $n^{-0,5}$ and $n^{-1}$ suggest that the integrands for these test cases have superposition dimensions $s_S$ greater than 1 and moderately large truncation dimension $s_T$. Another reason for these reduced rates can be the lack of smoothness in integrand [e.g., $I_t$ in (35)]. Integrand smoothness can lead to better QMC performance as indicated in [45], for instance.

The LHS convergence lines lie in between Monte Carlo and QMC as expected from our theoretical discussion. QMC shows lower variance than LHS across all test cases. If we assume that LHS is able to remove most of the estimate variance contributed by the 1-D part of the integrand, this observation suggests that these integrands have $s_T > 1$, as per our discussion in Section VI-C. Also, the significant improvement by LHS over Monte Carlo, at least for the bandgap and the SRAM cases, indicates that a large part of the integrand is purely 1-D. Overall, these results suggest that the effective dimension ($s_S$ or $s_T$) for these test cases are > 1, but small.

2) *Latent Unidimensionality and Variable Ordering:* Let us now try to quantify the contribution of the 1-D parts of

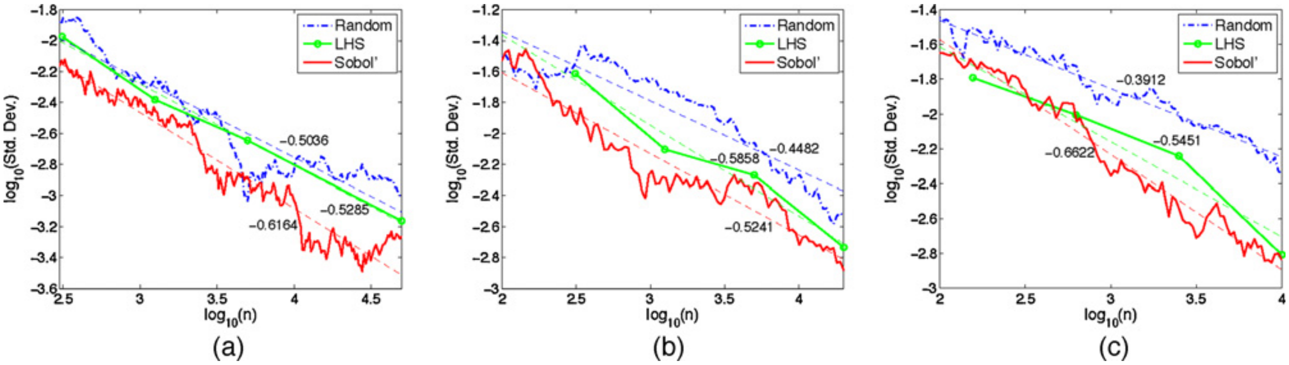

Fig. 9. Convergence of sample standard deviation of Monte Carlo (Random), LHS, and QMC (Sobol'). (a) MSFF yield convergence. (b) Bandgap yield convergence. (c) SRAM 90th percentile convergence.

TABLE I
ESTIMATED VARIANCE CONTRIBUTION FROM 1-D PART OF INTEGRAND

| | $\hat{\sigma}_{\text{MC}}$ | $\hat{\sigma}_{\text{LHS}}$ | $\widehat{\sigma_1^2/\sigma^2}$ |
|---|---|---|---|
| MSFF | 1.02$e$-3 | 6.86$e$-4 | 0.557 |
| Bandgap | 2.96$e$-3 | 1.84$e$-3 | 0.611 |
| SRAM column | 4.51e-3 | 1.56$e$-3 | 0.880 |

The "hat" denotes sample estimate.

the integrands. For this, we can use (28), where $\sigma_{\text{LHS}}$ and $\sigma_{\text{MC}}$ are estimated as the sample standard deviation across the 10 largest LHS runs and the 10 Monte Carlo runs, respectively. The results are tabulated in Table I. We see that more than half of the variance is contributed by the 1-D parts of the integrands. In particular, the SRAM column integrand is almost completely 1-D. This explains why LHS performs almost as well as QMC for this test case. The increasing unidimensionality of the integrands from the MSFF to the bandgap to the SRAM column matches with the observations in Figs. 8 and 7. The smaller the effective dimension of the integrand, the faster is the convergence expected from QMC on a properly order set of variables. Indeed, we do see that the effectiveness of the variable ordering matches the trend of unidimensionality of the integrand. Furthermore, since the MSFF case has a smaller dimensionality of $s = 31$, the difference in discrepancy between the higher and lower dimensions may not be as large as for the SRAM ($s = 403$). This further magnifies the effectiveness of the variable ordering for the SRAM as compared to the MSFF.

3) *Speedup Estimates:* Finally, we use the linear fits of Fig. 9 to estimate the sample sizes needed by Monte Carlo, LHS, and QMC to achieve a given level of accuracy. Say the exact value of the integral is $Q$ and we want the estimate $Q_n$ to be within $\delta$% of $Q$ with a confidence level of $p$. In other words, we want the probabilistic error magnitude to be less than or equal to $Q\left(\frac{\delta}{100}\right)$. Using the central limit theorem [25], we can write for the Monte Carlo estimate

$$\sigma_{\text{MC}}\Phi^{-1}\left(\frac{1+p}{2}\right) \leq Q\left(\frac{\delta}{100}\right) \tag{39}$$

where $\Phi$ is the standard normal CDF. Using the linear fits from Fig. 9, and the required $\sigma_{\text{MC}}$ computed from (39), we can estimate the sample size $\hat{n}$ required to satisfy this accuracy criterion. The same arguments hold for LHS and QMC. The results for $p = 95.45\%$ (corresponding to a $2\sigma$ level) and $\delta =$ 1 are shown in Table II. Since, we do not know the exact $Q$, we use an estimate computed using *all* the points at our disposal from the 10 Monte Carlo, 10 QMC, and the 10 largest LHS runs (see Fig. 8 for these estimates). We see significant speedups (2.4× to 8.6×) offered by QMC over Monte Carlo, showing the effectiveness of scrambled QMC as a variance reduction method. These speedups improve as the required accuracy increases. LHS also provides good speedup due to 1-D variance reduction, but QMC emerges the clear winner across all test cases.

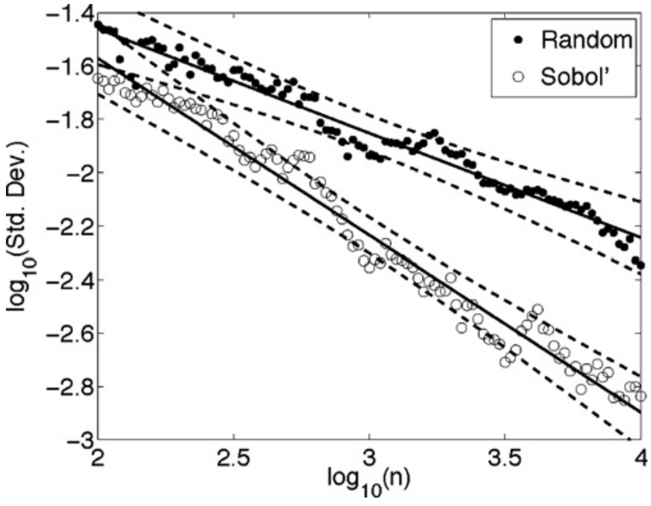

Fig. 10. 95% confidence interval bands for the convergence linear fits for the SRAM column.

Table II also gives the 95% confidence intervals (CI($\hat{n}$))) for the estimated sample sizes $\hat{n}$ for Monte Carlo and QMC. These confidence intervals are estimated from the least-square linear fits of Fig. 9 using standard linear regression theory [63], as follows. Let us say that the linear fits are given as

$$y = mx + c \tag{40}$$

computed from data $\mathbf{x} = \{x_1, \ldots, x_N\}$, $\mathbf{y} = \{y_1, \ldots, y_N\}$. In our case, $x \equiv \log_{10}(n)$ and $y \equiv \log_{10}(\sigma)$. Let us say we wish to compute the confidence intervals at confidence level $p$ ($p = 0.95$ for 95% confidence intervals). Let $t_{(1+p)/2,N-2}$ be the inverse CDF of the Student's $t$ distribution with $N-2$ degrees of freedom, at the cumulative probability value $(1+p)/2$. The

TABLE II

ESTIMATES OF NUMBER OF POINTS ($\hat{n}$) NEEDED TO ACHIEVE A GIVEN ERROR WITH A CONFIDENCE LEVEL OF 95.45%

| Case | | MC | | QMC | LHS | QMC/MC |
|---|---|---|---|---|---|---|
| | $\hat{n}$ | CI($\hat{n}$) | $\hat{n}$ | CI($\hat{n}$) | $\hat{n}$ | Speedup |
| MSFF | 1384 | [883, 1966] | 584 | [321, 897] | 1202 | 2.4× |
| Bandgap | 89 006 | [30 552, 629 198] | 10 349 | [5549, 28 805] | 16 623 | 8.6× |
| SRAM column | 1633 | [1097, 2671] | 354 | [191, 544] | 392 | 4.6× |

CI($\hat{n}$) are 95% confidence intervals for ($\hat{n}$).

residual or error variance in our linear regression is

$$\hat{\sigma}_\epsilon^2 = \frac{\sum_{i=1}^{N}(y_i - \bar{y})^2}{N-2}. \tag{41}$$

Then, the upper and lower bounds of the confidence interval of $x$ for some given $y^*$ can be computed by (numerically) solving

$$y^* - mx - c \pm t_{\frac{1+p}{2},N-2}\hat{\sigma}_\epsilon^2 \left( \frac{1}{N} + \frac{(x-\bar{x})^2}{\sum_{i=1}^{N}(x_i - \bar{(}x))^2} \right) = 0 \tag{42}$$

for $x$ [63]. Fig. 10 illustrates these confidence intervals using the SRAM column example. It shows the full 95% confidence interval bands for each linear fit.

The confidence intervals computed in Table II show minimal overlap, providing strong statistical support for the expected speedup from QMC over MC. Confidence intervals for LHS are not computed since the number of LHS data points ($N$) are very few: a consequence of the fact that the LHS point sets are of fixed predetermined lengths.

It should also be possible to apply other Monte Carlo variance reduction techniques [16], independently on top of QMC, to further improve accuracy, or achieve speedup gains.

## IX. CONCLUSION

Statistical problems from computational finance share key features with statistical problems in circuit analysis: they are essentially high-dimensional nonlinear integration problems which in practice exhibit lower latent dimensionality than the original integration domain. We demonstrated that one of the most celebrated techniques in the finance world, QMC simulation, can be successfully applied to statistical circuit yield problems, with attractive runtime speedups. However, this requires one to be careful in mapping these problems to the QMC form by exploiting their effective dimension. We showed how this can be done using appropriate sensitivity information. We further revealed the latent structure of circuit problems using the concept of ANOVA decomposition in the context of LHS. This analysis reveals the reasons for why QMC should perform better than both conventional Monte Carlo and LHS on our circuit problems, and indeed we observed this superiority of QMC in our simulation experiments.

## REFERENCES

[1] D. Hocevar, M. Lightner, and T. Trick, “A study of variance reduction techniques for estimating circuit yields,” *IEEE Trans. Comput.-Aided Design*, vol. 2, no. 3, pp. 279–287, Mar. 1983.

[2] M. Mani, A. Devgan, and M. Orshansky, “An efficient algorithm for statistical minimization of total power under timing yield constraints,” in *Proc. IEEE/ACM Des. Autom. Conf.*, 2005, pp. 309–314.

[3] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, “First-order incremental block-based statistical timing analysis,” in *Proc. IEEE/ACM Des. Autom. Conf.*, Jun. 2004, pp. 331–336.

[4] G. S. Fishman, *A First Course in Monte Carlo*. Pacific Grove, CA: Duxbury, 2006.

[5] K. Singhal and J. F. Pinel, “Statistical design centering and tolerancing using parameter sampling,” *IEEE Trans. Circuits Syst.*, vol. 28, no. 7, pp. 692–702, Jul. 1981.

[6] J. F. Swidzinski, M. Keramat, and K. Chang, “A novel approach to efficient yield estimation for microwave integrated circuits,” *IEEE Midwest Symp. Circuit Syst.*, vol. 1, no. 8, pp. 367–370, Aug. 1999.

[7] N. J. Elias, “Acceptance sampling: An efficient, accurate method for estimating and optimizing parametric yield,” *IEEE J. Solid-State Circuits*, vol. 29, no. 3, pp. 323–327, Mar. 1994.

[8] H. L. Abdel-Malik and A.-K. S. O. Hassan, “The ellipsoidal technique for design centering and region approximation,” *IEEE Trans. Comput.-Aided Design*, vol. 10, no. 8, pp. 1006–1014, Aug. 1991.

[9] K. J. Antreich, H. E. Graeb, and C. U. Weiser, “Circuit analysis and optimization driven by worst-case distances,” *IEEE Trans. Comput.-Aided Design*, vol. 13, no. 1, pp. 57–71, Jan. 1994.

[10] S. S. Sapatnekar, P. M. Vaidya, and S.-M. Kang, “Convexity-based algorithms for design centering,” *IEEE Trans. Comput.-Aided Design*, vol. 13, no. 12, pp. 1536–1549, Dec. 1994.

[11] A. Singhee and R. A. Rutenbar, “Beyond low-order statistical response surfaces: Latent variable regression for efficient, highly nonlinear fitting,” in *Proc. IEEE/ACM Des. Autom. Conf.*, 2007, pp. 256–261.

[12] T.-K. Yu, S. M. Kang, I. N. Hajj, and T. N. Trick, “Statistical performance modeling and parametric yield estimation of MOS VLSI,” *IEEE Trans. Comput.-Aided Design*, vol. 6, no. 6, pp. 1013–1022, Nov. 1987.

[13] P. Feldmann and S. W. Director, “Integrated circuit quality optimization using surface integrals,” *IEEE Trans. Comput.-Aided Design*, vol. 12, no. 12, pp. 1868–1879, Dec. 1993.

[14] X. Li, J. Le, L. T. Pileggi, and A. Stojwas, “Projection-based performance modeling for inter/intra-die variations,” in *Proc. IEEE/ACM Int. Conf. CAD*, 2005, pp. 721–727.

[15] S. Tezuka, *Uniform Random Numbers: Theory and Practice*. Boston, MA: Kluwer, 1995.

[16] P. Glasserman, *Monte Carlo Methods in Financial Engineering*. Berlin, Germany: Springer, 2004.

[17] J. G. Van der Corput, “Verteilungsfunktionen (in Dutch),” in *Proc. Ned. Akad. V. Wet.*, vol. 38. 1935, pp. 813–821.

[18] P. Acworth, M. Broadie, and P. Glasserman, “A comparison of some Monte Carlo and quasi-Monte Carlo techniques for option pricing,” in *Monte Carlo and Quasi-Monte Carlo Methods 1996*, H. Niederreiter, P. Hellekalek, G. Larcher, and P. Zinterhof, Eds. New York: Springer, 1998, pp. 1–18.

[19] S. Ninomiya and S. Tezuka, “Toward real-time pricing of complex financial derivatives,” *Appl. Math. Finance*, vol. 3, no. 1, pp. 1–20, 1996.

[20] A. Singhee and R. A. Rutenbar, “From finance to flip–flops: A study of fast quasi-Monte Carlo methods from computational finance applied to statistical circuit analysis,” in *Proc. Int. Symp. Quality Electron. Design*, 2007, pp. 685–692.

[21] A. Singhee, S. Singhal, and R. A. Rutenbar, “Practical, fast Monte Carlo statistical static timing analysis: Why and how,” in *Proc. IEEE/ACM Int. Conf. CAD*, 2008, pp. 190–195.

[22] V. Veetil, D. Sylvester, and D. Blaauw, “Efficient Monte Carlo based incremental statistical timing analysis,” in *Proc. IEEE/ACM Des. Autom. Conf.*, 2008, pp. 676–681.

[23] J. Jaffari and M. Anis, “On efficient Monte Carlo-based statistical static timing analysis of digital circuits,” in *Proc. IEEE/ACM Int. Conf. CAD*, 2008, pp. 196–203.

[24] A. Singhee, “Novel algorithms for fast statistical analysis of scaled circuits,” Ph.D. dissertation, Dept. Electr. Comput. Eng., Carnegie Mellon Univ., Pittsburgh, PA, 2007.

[25] R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, 3rd ed. New York: Macmillan, 1971.

[26] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*. Philadelphia, PA: SIAM, 1992.

[27] F. J. Hickernell, "A generalized discrepancy and quadrature error bound," *Math. Comp.*, vol. 67, no. 221, pp. 299–322, 1998.

[28] E. Hlawka, "Functionen von beschränkter variation in der theori der gleichverteilung (in German)," *Annali di Matematica Pura ed Applicata*, vol. 54, no. 1, pp. 325–333, Dec. 1961.

[29] H. Niederreiter, "Quasi-Monte Carlo methods and pseudo-random numbers," *Bull. Am. Math. Soc.*, vol. 84, no. 6, pp. 957–1041, 1978.

[30] R. E. Caflisch, W. Morokoff, and A. Owen, "Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension," *J. Comp. Finance*, vol. 1, no. 1, pp. 27–46, 1997.

[31] J. Kiefer, "On large deviations of the empirical d.f. of vector chance variables and a law of the iterated logarithm," *Pacific J. Math.*, vol. 11, no. 2, pp. 649–660, 1961.

[32] I. M. Sobol', "The distribution of points in a cube and the approximate evaluation of integrals (English translation)," *U.S.S.R. Comp. Math. Math. Phys.*, vol. 7, no. 4, pp. 86–112, 1967.

[33] J. H. Halton, "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals," *Numer. Math.*, vol. 2, no. 1, pp. 84–90, 1960.

[34] J. M. Hammersley, "Monte Carlo methods for solving multivariate problems," *Ann. New York Acad. Sci.*, vol. 86, no. 3, pp. 844–874, 1960.

[35] H. Faure, "Discrépance de suites associées à un système de numération (en dimension s) (in French)," *Acta Arithmetica*, vol. 41, no. 1, pp. 337–351, 1982.

[36] H. Niederreiter, "Low-discrepancy and low-dispersion sequences," *J. Number Theory*, vol. 30, no. 1, pp. 51–70, Sep. 1988.

[37] C. P. Xing and H. Niederreiter, "A construction of low-discrepancy sequences using global function fields," *Acta Arith.*, vol. 73, no. 1, pp. 87–102, 1995.

[38] F. J. Hickernell, H. S. Hong, P. L'Ecuyer, and C. Lemieux, "Extensible lattice sequences for quasi-Monte Carlo quadrature," *SIAM J. Sci. Comp.*, vol. 22, no. 3, pp. 1117–1138, 2000.

[39] L. K. Hua and Y. Wang, *Applications of Number Theory to Numerical Analysis*. Berlin, Germany: Springer, 1981.

[40] P. Bratley and B. L. Fox, "Algorithm 659: Implementing Sobol's quasirandom sequence generator," *ACM Trans. Math. Soft.*, vol. 14, no. 1, pp. 88–100, 1988.

[41] S. Joe and F. Y. Kuo, "Remark on algorithm 659: Implementing Sobol's quasirandom sequence generator," *ACM Trans. Math. Soft.*, vol. 29, no. 1, pp. 49–57, 2003.

[42] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2nd ed. Cambridge, MA: MIT Press, 1972.

[43] J. Rifà and J. Borrell, "A fast algorithm to compute irreducible and primitive polynomials in finite fields," *Theory Comput. Syst.*, vol. 28, no. 1, pp. 13–20, 1995.

[44] I. A. Antonov and V. M. Saleev, "An economic method of computing $LP_\tau$-sequences (English translation)," *U.S.S.R. Comp. Math. Math. Phys.*, vol. 19, no. 1, pp. 252–256, 1979.

[45] B. Moskowitz and R. E. Caflisch, "Smoothness and dimension reduction in quasi-Monte Carlo methods," *Mathl. Comput. Modelling*, vol. 23, nos. 8–9, pp. 37–54, 1996.

[46] X. Wang and K.-T. Fang, "The effective dimension and quasi-Monte Carlo integration," *J. Complexity*, vol. 19, no. 2, pp. 101–124, 2003.

[47] W. H. Press, B. P. Flannery, A. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge, MA: Cambridge University Press, 1992.

[48] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[49] M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.

[50] A. B. Owen, "Scrambled net variance for integrals of smooth functions," *Ann. Stats.*, vol. 25, no. 4, pp. 1541–1562, 1997.

[51] X. Wang and I. H. Sloan, "Low discrepancy sequences in high dimensions: How well are their projections distributed?" *J. Comput. Appl. Math.*, vol. 213, no. 2, pp. 366–386, 2007

[52] A. B. Owen, "Randomly permuted (t, m, s)-nets and (t, s)-sequences," in *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, H. Niederreiter and P. J.-S. Shiue, Eds. New York: Springer, 1995, pp. 299–317.

[53] A. B. Owen, "Variance with alternative scramblings of digital nets," *ACM Trans. Modeling Comp. Sim.*, vol. 13, no. 4, pp. 363–378, 2003.

[54] H. S. Hong and F. J. Hickernell, "Algorithm 823: Implementing scrambled digital sequences," *ACM Trans. Math. Soft.*, vol. 29, no. 2, pp. 95–109, 2003.

[55] G. Ökten and W. Eastman, "Randomized quasi-Monte Carlo methods in pricing securities," *J. Econ. Dyn. Control*, vol. 28, no. 12, pp. 2399–2426, 2004.

[56] M. Matsumoto and Y. Kurita, "Twisted GFSR generators II," *ACM Trans. Modeling Comp. Sys.*, vol. 4, no. 3, pp. 254–266, 1994.

[57] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," *Ann. Math. Stats.*, vol. 29, no. 2, pp. 610–611, 1958.

[58] P. J. Acklam. *An Algorithm for Computing the Inverse Normal Cumulative Distribution Function* [Online]. Available: http://home.online.no/pjacklam/notes/invnorm

[59] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45 nm early design exploration," *IEEE Trans. Electron Devices*, vol. 53, no. 11, pp. 2816–2823, Nov. 2006.

[60] E. Hewitt and K. Stromberg, *Real and Abstract Analysis*. Berlin, Germany: Springer, 1965.

[61] H. Banba, H. Shiga, A. Umezawa, and T. Miyaba, "A CMOS bandgap reference circuit with sub-1-V operation," *IEEE J. Solid-State Circuits*, vol. 34, no. 5, pp. 670–674, May 1999.

[62] P. R. Gray, P. J. Jurst, S. H. Lewis, and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 4th ed. New York: Wiley, 2001.

[63] F. A. Graybill, *Theory and Application of the Linear Model*. Pacific Grove, CA: Duxbury, 1976.

**Amith Singhee** (S'06–M'09) received the B.Tech. (Hons.) degree in electrical engineering from the Indian Institute of Technology (IIT) Kharagpur, Kharagpur, India, in 2000, and the M.S. and Ph.D. degrees, both in electrical and computer engineering, from Carnegie Mellon University, Pittsburgh, PA, in 2002 and 2007, respectively.

He was with Neolinear, Inc., Pittsburgh, and subsequently with Cadence Design Systems, Inc., San Jose, CA, from 2002 to 2004. He is currently a Research Staff Member with the IBM T. J. Watson Research Center, Yorktown Heights, NY. His current research interests include statistical simulation, process variation modeling, and general design for manufacturability.

Dr. Singhee is a recipient of several awards, including the European Design Automation Association Outstanding Dissertation Award, the A. G. Milnes Award for his Ph.D. dissertation, the Silver Medal at IIT Kharagpur, the Best Paper Award at the Design Automation Conference in 2002 and the Design, Automation and Test in Europe in 2007, and the Best Student Paper Award at the International Conference on Very Large Scale Integration Design in 2008. His paper on memory statistics was published in the book *The Most Influential Papers of 10 Years DATE*, and he has received multiple Inventor Recognition Awards from the Global Research Consortium.

**Rob A. Rutenbar** (S'77–M'84–SM'90–F'98) received the Ph.D. degree from the University of Michigan, Ann Arbor, in 1984.

From 1984 to 2009, he was a Faculty Member with Carnegie Mellon University, Pittsburgh, PA, where he held the Stephen J. Jatras (E'47) Chair in Electrical and Computer Engineering. In 2010, he joined the University of Illinois at Urbana-Champaign, Urbana, where he is currently the Abel Bliss Professor and Head of Computer Science. He has worked on tools for custom circuit synthesis and optimization for over 25 years. He has published over 150 papers throughout his career, and his work has been featured in venues ranging from *EE Times* to *Economist Magazine*.

Dr. Rutenbar has received many awards throughout his career. He has received several Best Paper Awards at the Design Automation Conferences 1987, 2002, and 2010, Design, Automation and Test in Europe 2007, and the International Conference on VLSI Design 2008. He was the 2001 winner of the Semiconductor Research Corporation Aristotle Award for Excellence in Education and the 2007 winner of the IEEE Circuits and Systems Industrial Pioneer Award for his work in making analog synthesis a commercial technology. He is a Fellow of the ACM.