From Finance to Flip Flops: A Study of Fast Quasi-Monte Carlo Methods from Computational Finance Applied to Statistical Circuit Analysis

Amith Singhee, Rob A. Rutenbar
Dept. of ECE, Carnegie Mellon University, Pittsburgh, Pennsylvania, 15213 USA
{asinghee,rutenbar}@ece.cmu.edu

Abstract

Problems in computational finance share many of the characteristics that challenge us in statistical circuit analysis: high dimensionality, profound nonlinearity, stringent accuracy requirements, and expensive sample simulation. We offer a detailed experimental study of how one celebrated technique from this domain -- Quasi-Monte Carlo (QMC) analysis -- can be used for fast statistical circuit analysis. In contrast with traditional pseudo-random Monte Carlo sampling, QMC substitutes a (shorter) sequence of deterministically chosen sample points. Across a set of digital and analog circuits, in 90nm and 45nm technologies, varying in size from 30 to 400 devices, we obtain speedups in parametric yield estimation from 2X to 50X.

1. Introduction

Continued device scaling has dramatically increased the statistical variability with which tomorrow's circuits must contend. In a few special cases, we have analytical methods that can give us the deterministic answers we seek, e.g., optimal sizing and threshold assignment in combinational logic under statistical yield and timing constraints [1]. Unfortunately, such analytical solutions remain rare. In the general case, some combination of complex statistics, high dimensionality, profound nonlinearity or non-normality, stringent accuracy, and expensive performance evaluation (e.g., detailed simulation) thwart our analytical aspirations. What remains are the Monte Carlo methods [2].

The first contribution of this paper is the simple observation that there is *another* application domain characterized by many of the same technical challenges we face with semiconductors. That domain is *computational finance*. Indeed, the parallels are striking. There are celebrated analytical results, for example, the Nobel Prize winning Black-Scholes model for option pricing [2]. But there is also the reality that, as financial instruments have become ever more complex and subtle, analytical models have given way to Monte Carlo as the only practical analysis method [2]. The problems are not only very nonlinear, they can also be quite large: pricing a portfolio of options or securities over a several year horizon can create problems with 1000+ statistical variables [3]. Accuracy is often required to the level of one basis point (a relative accuracy of 10^{-4}) under impressively short time constraints (minutes, in the case of real-time arbitrage).

The natural question becomes: can we *redeploy* any of these methods, moving them from finance to flip flops? In particular, can we retarget recent Monte Carlo methods developed for quickly pricing complex financial instruments, to the problem of estimating statistical quantities of interest in deeply scaled circuits? To be concrete: does the deep statistical structure of pricing a 30-year mortgage backed security resemble, in any practical and exploitable way, the structure of random dopant fluctuations in an SRAM column?

As it turns out, the answer is "yes". In this paper we offer a rigorous experimental study of one of the most celebrated methods developed in computational finance in the last decade: the *Quasi Monte Carlo* (QMC) method. As with all Monte Carlo (MC) methods, the goal is to converge to the required accuracy as rapidly as possible, with as few sample simulations as possible. Although the underpinnings of QMC are not new [4], recent improvements in both theory and implementation complexity, along with the empirical discovery that these methods are unexpectedly efficient at high-dimensional statistical integral evaluation, propelled these techniques onto center stage in the computational finance world [5].

Unfortunately, like all complex mathematical methods, correct application requires adapting the strengths of the methods to the specifics of the problem. In other words, we cannot apply these ideas blindly and expect to extract maximum (or, perhaps, any) benefit. In the remainder of the paper, we review the convergence theory for both standard MC and QMC methods, and show how to correctly apply these ideas to a range of statistical circuit analysis problems.

2. Standard Monte Carlo Convergence Behavior

MC methods are typically used to approximate some integral of the following standard form:

$$I(f) = \int_{C^s} f(\boldsymbol{x}) d\boldsymbol{x}\,,\ \boldsymbol{x} = (x_1, \ldots, x_s) \tag{1}$$

where $C^s = [0,1)^s$ is the s-dimensional unit cube, and f is some integrable function. The MC approximation is given by

$$Q(f) = n^{-1} \sum_{i=1}^{n} f(\boldsymbol{x}_i)\,, \tag{2}$$

where $\boldsymbol{x}_i$ are n independent and identically distributed samples drawn from the s-dimensional uniform distribution $U[0,1)^s$. Problems with different variable ranges, arbitrary statistical distributions, arbitrary nonlinearity, etc., can always be transformed into this canonical integral form, i.e., these can always be included in our function f, without any loss of generality. Thus, the problems we discuss are all defined over the s-dimensional unit cube. Parametric yield computation for circuits also follows the form in (1). Given this, let us look at the convergence properties of standard MC.

If f has finite variance

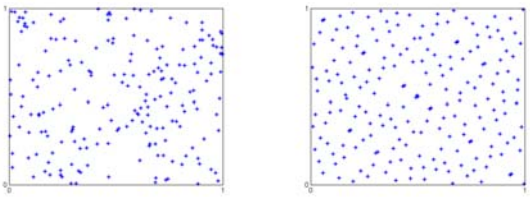
$$\sigma^2(f) = \int_{C^s} [f(\boldsymbol{x}) - I(f)]^2 dx \tag{3}$$

the mean square error of the MC integral approximation is given as

$$E[(Q(f) - I(f))^2] = \sigma^2(f)/n \tag{4}$$

Hence, the expected MC error is $O(n^{-1/2})$. The advantage of standard MC is that this error does not depend on the dimensionality s.

There is another way to look at the error, using the concept of *discrepancy*. Fig. 1 shows 200 uniformly distributed pseudo-random points in C^2. The points are, indeed, uniformly distributed, but geometrically, they are not equally separated. We can see that the points exhibit both clumps and empty holes. Discrepancy is a quantity used to reflect this geometric *non*-uniformity of points in a set.



Pseudo-random (200) Quasi-random (200)

FIGURE 1. Pseudo-random points exhibiting higher discrepancy than a low-discrepancy quasi-random sequence.

There are several definitions of discrepancy [6], the simplest being the *Star Discrepancy*, or the L_∞ *-discrepancy*:

$$D^*_n = \sup_J |A(J;n)/n - Vol(J)|, \quad (5)$$

where $J \in C^s$ is any s-dimensional hyper-rectangle with one corner at 0. $Vol(J)$ is the volume of J, and $A(J;n)$ is the number of points inside J. Geometrically speaking, the star discrepancy measures how well the (relative) volume of any origin-anchored hyper-rectangle in the unit cube is approximated by the fraction of sample points that lie in that volume. Surprisingly enough, samples from the standard uniform distribution $\boldsymbol{x}_i \sim U[0,1)^s$ may show extremely large discrepancy, as Fig. 1 so clearly illustrates.

The Koksma-Hlawka theorem [7] quantifies this effect. If f has a suitably bounded variation $V(f)$ then the absolute integration error is itself bounded by the star discrepancy, as:

$$|Q(f) - I(f)| \le V(f) D^*_n \quad (6)$$

($V(f)$ itself has a rather technical definition; see [8].) The larger implication is that sample points with *lower* discrepancy can produce integral estimates with *lower* errors.

The first obvious question is: what is the discrepancy for standard Monte Carlo? Chung [9] showed that, for uniform points $\boldsymbol{x}_i \sim U[0,1)^s$,

$$D^*_n = O\left(\left(\frac{\log\log n}{n}\right)^{1/2}\right) \quad (7)$$

Thus, we see an echo of the familiar $1/\sqrt{n}$ convergence behavior. But the real question is this: are there sampling sequences that guarantee a *better*, lower discrepancy? The answer is "yes".

3. Quasi-Monte Carlo

Sequences with asymptotically superior discrepancy exist and are known as *Low Discrepancy Sequences* (LDSs). Fig. 1 also shows 200 points drawn from such a sequence (points from a so-called *Sobol'* sequence[10]). The higher uniformity, as compared to pseudo-random points, is obvious. Theoretically, the discrepancy bound for these points is

$$D^*_n = O((\log n)^s / n) \quad (8)$$

and they possess the surprising attribute that they are generated *deterministically*, in contrast to the standard pseudo-random sampling of classical MC. Monte Carlo performed using samples generated deterministically from a low discrepancy sequence is known as *Quasi-Monte Carlo* (QMC). LDSs are also known as *Quasi-Random Sequences*. The overall idea is conceptually simple: rather than randomly sampling the space, we try to "fill the space" with samples that are as geometrically, homogeneously equidistant as possible.

Comparing the bounds of (7) and (8) gives us some sense of the possible advantages, and challenges, of the method. Comparing denominators, we see the tantalizing possibility of *linear* convergence for QMC. 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 empirical situations, this turns out not to be the case; we shall return to this in Section 5.

The first construction of an LDS for all problem dimensions s was given by Halton in 1960 [4]. Other constructions have been introduced by Sobol' [10], Faure [11], Niederreiter [12] and Niederreiter and Xing (NX) [13]. Space does not permit any detailed survey of the different strategies here; see [2] for a survey. Niederreiter showed a general construction principle for one large and popular class of LDSs called *(t,s)-sequences*[6]. We use one particularly successful set of (t,s)-sequence, called Sobol' point, for our experiments.

4. Sobol' Points

Sobol's construction, introduced in [10], is one of the most popular in current use. Sobol' points perform significantly better than the original Halton points in terms of discrepancy. Also, empirical results [2][14] suggest that Sobol' points perform better than Faure points—at least, for modern computational finance applications. The NX points promise to have significantly better discrepancy [13]. However, their implementation is significantly more complex and, currently, not flexible enough for an arbitrary problem dimension s, requiring the solution of a set of thorny number theoretic problems for each dimension. For all these reasons, we choose the Sobol' points as our representative LDS. Let us briefly describe their construction.

We use the implementations in [15] and [16]. First, suppose we are working in just one dimension, i.e., $s = 1$. We choose *one* primitive polynomial [17] in the field $\boldsymbol{Z}_2$ (coefficients from {0,1})

$$P \equiv x^d + a_1 x^{d-1} + \ldots + a_{d-1} x + 1 \quad (9)$$

We also choose odd integers $m_1, \ldots, m_d$, such that $0 < m_j < 2^j$. Define *direction numbers*

$$v_j = m_j / 2^j, \; j \le d \quad (10)$$

and their recurrence relation (in Boolean operations)

$$v_j = a_j v_{j-1} \oplus \ldots \oplus a_{d-1} v_{j-d-1} \oplus v_{j-d} \oplus (v_{j-d}/2^d), \; j > d \quad (11)$$

This gives us a set of direction numbers v_j for $j > 0$. To compute the n-th Sobol' value x_n, we use

$$x_n = n_1 v_1 \oplus n_2 v_2 \oplus \ldots, \quad (12)$$

where $\ldots n_3 n_2 n_1$ is the *Gray code* representation of n. Using the Gray code representation is must faster than using the binary representation, since only one bit changes in the Gray code from n to $n+1$, making the operation (12) incremental (only one XOR). This reshuffling does not affect the asymptotic discrepancy.

For a general problem with $s > 1$ dimensions, we now choose s *different* primitive polynomials and generate sequences for *each* coordinate, using the above method. The polynomials are chosen sequentially with non-decreasing degree d, for increasing dimension.

One additional problem is how to choose the initial m_j values for each dimension i. Let us name these as $m_{i,j}$. Also, renaming the direction numbers as $v_{i,j}$, where i is the dimension $1 \le i \le s$, we define $v_{i,j,1}$ as the first bit after the binary point of $v_{i,j}$. Set

$$V_d = [v_{i,j,1}], \text{ where } 1 \le i \le d \text{ and } 1 \le j \le d \quad (13)$$

Then, according to Sobol's development in [10], the condition

$$det(V_d) = 1 (mod\ 2) \quad (14)$$

gives better uniformity. Hence, $m_{i,j}$ are chosen to satisfy (14) (see [16]).

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. However, all LDS schemes suffer from some idiosyncrasies when applied to higher dimensional problems, requiring additional finesse in the way we map our statistical integration problems into a viable QMC formulation. We treat these problems next.

5. Minimizing QMC Integration Error: Effective Dimension & LDS Pattern Effects

Looking only at the asymptotics, the $O((\log n)^s/n)$ error bound of QMC should show *no* runtime improvements over the $O(n^{-0.5})$ bound of conventional MC for very large s and feasibly large n. However, QMC has been seen to outperform MC even for problems with very large s, e.g., IBM's 1439-dimensional derivative-pricing experiments of [3]. This anomalous, empirical success has been largely explained using the concept of *effective dimension* [8]. We review the concept here, because it strongly impacts the manner in which will map our circuits problems into a successful QMC form.

Let us first review the concept of the *Analysis of Variance (ANOVA) Decomposition.* The decomposition expresses a function $f(\boldsymbol{x})$ as a sum of simpler functions $f_u(\boldsymbol{x})$, each depending on a subset of the inputs $\boldsymbol{x} = (x_1, \ldots, x_s)$. For any subset $u \subseteq \{1, \ldots, s\}$, let $-u$ be its complementary set $\{1, \ldots, s\} - u$ and let $\boldsymbol{x}_u = \{x_i\}, i \in u$ be the sub-vector of coordinates of $\boldsymbol{x}$ corresponding to u. Also, let C^u denote the unit cube in the dimensions that belong to u. Then, for any square integrable function f, the ANOVA decomposition is

$$f(\boldsymbol{x}) = \sum_{u \subseteq \{1, \ldots, s\}} f_u(\boldsymbol{x}), \quad (15)$$

where the ANOVA terms follow the recursion

$$f_u(\boldsymbol{x}) = \int_{C^{-u}} f(\boldsymbol{x}) d\boldsymbol{x}_{-u} - \sum_{v \subset u} f_v(\boldsymbol{x}). \quad (16)$$

and are orthogonal. Hence, the variance of f can be written as

$$\sigma^2(f) = \sum_{|u|>0} \sigma_u^2(f), \text{ where } \sigma_u^2(f) = \int_{C^s} (f_u(\boldsymbol{x}))^2 d\boldsymbol{x} \quad (17)$$

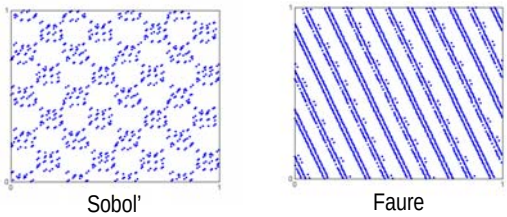
Then, taking p as a selected probability value close to 1, we have the following definitions.

Definition 1. The *effective dimension* of f, in the *superposition* sense, is the smallest integer s_S, s.t. $\sum_{0<|u|\le s_S} \sigma_u^2(f) \ge p\sigma^2(f)$.

Definition 2. The *effective dimension* of f, in the *truncation* sense, is the smallest integer s_T, s.t. $\sum_{u \subseteq \{1, \ldots, s_T\}} \sigma_u^2(f) \ge p\sigma^2(f)$.

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

Effective dimension is relevant to us for two important reasons. First, it is widely invoked to help explain why QMC has been so strikingly efficient (e.g., 150X speedup[3]) on large financial problems. These tasks seem to have low effective dimension; for example, in a pricing task with a long time horizon. money today is much more valuable than money tomorrow, which reduces the impact of many dimensions of the problem. It is an open question if this behavior obtains in circuit analysis. Second, effective dimension is essential to optimally map problems into QMC form, which we discuss next.



Sobol' Faure

FIGURE 2. Undesirable patterns in 2-D projections of low-discrepancy sequences.

Ideally, it should not matter how we assign problem variables to elements in our LDS points $\boldsymbol{x} = (x_1, \ldots, x_s)$. Suppose we have, say, 100 random threshold voltages to sample. It should not matter if any particular $V_t(i)$ voltage is mapped to x_1, or x_{37}, or x_{99}. Unfortunately, this is not the case. All LDSs are imperfect, and usually show degraded uniformity as dimension increases. This takes the form of *pattern dependencies* [8], illustrated in Fig. 2. If we take two arbitrary elements (x_1, x_j) from point $\boldsymbol{x} = (x_1, \ldots, x_s)$, we should expect to see a low-discrepancy 2-D projection such as Fig. 1. This is not always the case, as shown for two LDSs in Fig. 2.

We can finesse this problem by trying to assign the most "important" statistical variables to the lower, less pattern sensitive coordinates of x. More formally, in the language of ANOVA, we can write

$$|Q(f) - I(f)| \le \sum_{u \subseteq \{1, \ldots, s\}} V_u(f_u) D^*_{n,u}, \quad (18)$$

where $V_u(f_u)$ is the variation of f_u taken as a $|u|$-dimensional function [18], and $D^*_{n,u}$ is the star-discrepancy of the $|u|$-dimensional points obtained by projecting the sequence onto the coordinates in u. This suggests that if f has low s_S, then, because of lower $D^*_{n,u}$, QMC will perform better than MC. but to deal with the pattern effects of Fig. 2. we should map the input values $\{x_i\}$ to the coordinates of the LDS such that subsets u with large σ_u $(V_u(f_u))$ and small $D^*_{n,u}$ coincide, and those with large $D^*_{n,u}$ and small σ_u coincide. If we can do this, we can still achieve very low error.

For problems with a time-series random-walk structure, there are good techniques for mapping [19], but these are not applicable in the case of circuit yield analysis. Principal Components analysis (PCA) is obviously useful, but even here, we still need to be able to best map the problem to a QMC form after PCA has completed. We can suggest two strategies:

1. The designer selects the parameters that most affect the relevant performance metrics, and assigns these to the lower coordinates of the QMC.

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 concentrate on the latter method here. The measure of sensitivity that we use is the absolute value of the *Spearman's Rank Correlation Coefficient* [20]. This is similar to Pearson's Correlation, but more robust in the presence of non-linear relationships. Suppose R_i and S_i are the ranks of corresponding values of a parameter and a metric, then their rank correlation is given as:

$$r_S = \frac{\sum_i (R_i - \bar{R})(S_i - \bar{S})}{\sqrt{\sum_i (R_i - \bar{R})^2} \sqrt{\sum_i (S_i - \bar{S})^2}} \quad (19)$$

This approach has a two-fold advantage. First, it helps reduce the truncation dimension, since all the important dimensions are the first few. Second, the first few dimensions of the Sobol' points are more uniform for small n [2][19], and this approach helps map the important subset of variables (large σ_u) to the dimensions with good uniformity (small $D^*_{n,u}$). The rank correlation can be computed by first running a smaller MC run. For multiple metrics, we use the sum of the rank correlation values across all the metrics.

6. Randomized QMC

We now confront one final problem: the error bound (6) for QMC is very difficult to compute. Also, it is only an upper bound on the error: it does not provide a practical way to *measure* the actual error, if the exact solution is unknown. In a standard MC scenario, we would simply run several different pseudo-random samplings, and compare them. But QMC generates deterministic samples: each run yields the *same* samples. To address this, Owen [21] introduced *Randomized QMC* (RQMC) to estimate the variance, using so-called *scrambled* versions of the same LDS. Let $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\}$ and $\{\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots\}$ denote the original LDS and a randomly scrambled version, respectively. Let $x_{ni} = 0.x_{ni1}x_{ni2}\ldots$ be the i-th coordinate of $\boldsymbol{x}_n$. Then,

$$y_{ni1} = \pi_i(x_{ni1}), \text{ and } y_{nik} = \pi_{x_{ni1}, \ldots, x_{nik-1}}(x_{nik}) \text{ for } k > 1 \quad (20)$$

where $\pi_{(\ldots)}$ are random permutations of $\{0, 1, \ldots, b-1\}$, chosen uniformly and mutually independently. Hence, this method scrambles the digits of the original LDS. Other methods have also been introduced [22]. All these randomized sequences maintain the uniformity properties of the original LDS.

Owen's original scrambling uses a large amount of memory. Hence, we use a more scalable, but less powerful, version described in [23].

7. Experimental Results for Circuit Analysis

In this section, we compare the performance of the scrambled Sobol' points against the performance of standard Monte Carlo, on three different testcases. First, we make some observations about our MC and RQMC implementations:

- Since LDSs perform better when the first few points are skipped, we skip the first $n_{skip} = 2^{\lfloor log_2 n \rfloor}$ Sobol' points [14].
- A Linear Congruential Generator (LCG) [2] (drand48() in C) was used to generate the pseudo-random sequences for standard MC because of its widespread popularity. Variance results in [24], comparing LCG with a Generalized Feedback Shift Register Generator (GFSR) [25], do not show significant improvement for GFSR, relative to the improvement with RQMC.
- The standard Box Muller method for generating normally distributed variates is inaccurate, especially for a large number of samples [26]. Hence, an inverse transform method was used.

Now, we describe the testcases and the experiments. All samples were evaluated using detailed circuit simulation in Cadence Spectre. Results for all testcases will be analyzed together in Section 7.4.

7.1 Master-Slave Flip-Flip with Scan (MSFF)

The first testcase is a commonly seen Master-Slave Flip-Flop with scan chain (MSFF) Fig. 3. The design has been implemented using the 45 nm CMOS Predictive Technology Models of [27]. Variations considered are Random Dopant Fluctuation (RDF) for all transistors and one global gate-oxide (t_{ox}) variation The RDF is

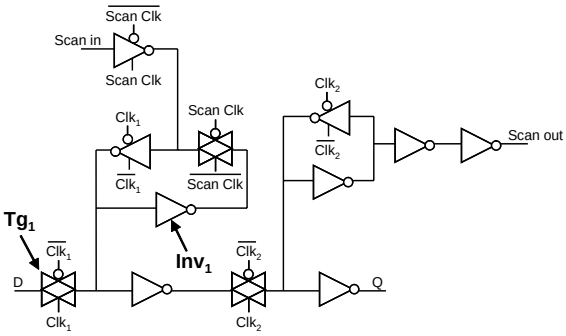


FIGURE 3. Master-Slave Flip-Flop with Scan Chain component.

modeled as normally distributed threshold voltage (V_t) variation:

$$\sigma(V_t) = 0.0135 V_{t0} / \sqrt{WL} \text{ where W,L are in } \mu\text{m} \quad (21)$$

V_{t0} is the nominal threshold voltage. This results in 30% standard deviation for a minimum-sized transistor. The t_{ox} standard deviation is taken as 2%. The metric being measured is the clock-output delay, τ_{cq}. The integral being estimated is the parametric yield, with a maximum acceptable delay of τ_{max} = 200ps. If we define

$$I_t(\boldsymbol{z}, f(\boldsymbol{z})) = \begin{cases} 1 & , f(\boldsymbol{z}) \le t \\ 0 & , f(\boldsymbol{z}) > t \end{cases} \quad (22)$$

We can express yield in the form (1) as follows:

$$Y_t(f, \Phi) = \int_{C^s} I_t(\Phi(\boldsymbol{x}), f(\Phi(\boldsymbol{x})))d\boldsymbol{x} \quad (23)$$

where $\Phi(\boldsymbol{x})$ transforms uniformly distributed $\boldsymbol{x} \in [0, 1)^s$ to the required joint distribution (normal in this case). There are a total of 31 statistical variables in this problem. For the MSFF, yield will be given as $Y_{\tau_{max}}(\tau_{cq}, \Phi)$. 10 MC runs of 50,000 pseudo-random points each were run. 1 QMC run with 50,000 Sobol' points, and 9 QMC runs with 50,000 scrambled Sobol' points each were also run. Results are discussed in Section 7.4.

As an illustrating example, let us look at how the rank correlation-based variable-dimension mapping works for this testcase. Fig. 4 shows the absolute value of the rank correlation ($|r_S|$) of each circuit parameter with the clock-output delay, for rising output, computed from an initial MC run of 1000 samples. The variable are sorted according to decreasing importance ($|r_S|$): in the order they would be mapped to the 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 of the pMOS device in the input transmission gate Tg_1, and 3) N_{inv1}: the V_t variation of the nMOS device in the inverter Inv_1. The latter two devices are on the critical signal path for a high input causing a rising output, and are

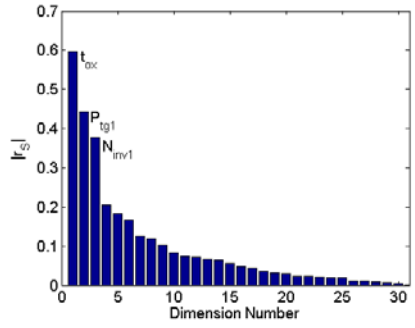


FIGURE 4. MSFF parameters sorted by decreasing importance, estimated by the absolute value of rank correlation. The most important parameters are mapped to the earlier dimensions.

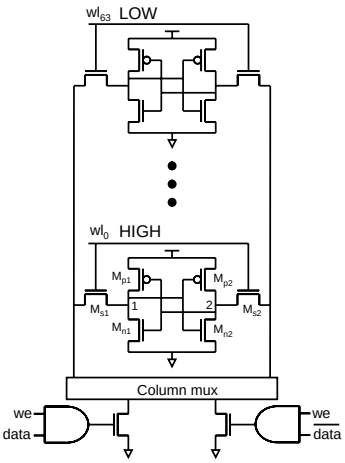


FIGURE 5. 64-bit SRAM column with column mux and write drivers. V_t variation on all devices and global t_{ox} variation.

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.

7.2 64-bit SRAM Column

Yield analysis of SRAMs is unavoidable, given the large capacity of SRAMs and large variation due to RDF. Our second testcase is a 64-bit SRAM Column, with non-restoring write driver and column multiplexor (Fig. 5). Only one cell is being accessed, while all the other wordlines are turned off. The device models used are from the Cadence 90nm Generic PDK library. RDF on all 402 devices (including the write driver and column mux) are considered, along with one global gate-oxide variation. All variations are assumed to be normally distributed. The V_t standard deviation is taken as

$$\sigma(V_t) = 5mV/\sqrt{WL} \text{ where W,L are in } \mu\text{m} \qquad (24)$$

This variation is too large for the 90nm process, but is in the expected range for more scaled technologies. $\sigma(t_{ox})$ is taken to be 2%.

The metric being measured is the write time τ_w: the time between the wordline going high to the non-driven cell node (node 2) transitioning. Here, "going high" and "transitioning" imply crossing 50% of the full voltage change. The write time is measured as a multiple of the fanout-4 delay of an inverter (FO4). The value being estimated is the 90-th percentile of the write time. If we write

$$J_p(\mathbf{z}, Y_t, f, \Phi) = \begin{cases} 0 & , Y_{f(\Phi(\mathbf{z}))}(f, \Phi) \neq p/100 \\ 1 & , Y_{f(\Phi(\mathbf{z}))}(f, \Phi) = p/100 \end{cases}, \qquad (25)$$

any p-th percentile can be expressed in form (1) as

$$\pi_p(f, \Phi) = \int_{C^S} J_p(\mathbf{x}, Y_t, f, \Phi) d\mathbf{x} . \qquad (26)$$

Then, the 90-th percentile in this case will be $\pi_{90}(\tau_w, \Phi)$.

10 MC runs of 20,000 pseudo-random points each were run. 1 QMC run of 20,000 Sobol' points and 9 QMC runs of 20,000 scrambled Sobol' points each, were also run. Results are discussed in Section 7.4.

7.3 Low-Voltage CMOS Bandgap Reference

Fig. 6 shows a low-voltage CMOS Bandgap Reference circuit [28]. This bandgap is able to provide reference voltages that are less than 1 Volt, and is built using standard CMOS technology. This circuit was chosen for its relevance in today's low-voltage designs, and

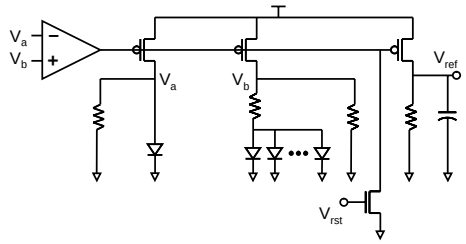


FIGURE 6. A 0.6-V-output CMOS bandgap

also to test QMC on a circuit with highly non-linear behavior. The opamp used is a standard single-ended RC-compensated two-stage opamp [29]. The circuit has 101 diodes. The transistor device and variation models are the same 90nm CMOS as the SRAM. RDF in the diodes is modeled as normally distributed variations 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_{ref}), 2) settling time (τ_s) and 3) dropout voltage (V_{do}). V_{do} is the difference between the supply voltage and V_{ref} when V_{ref} falls by 1% of its nominal value (0.6V): lower V_{do} implies a more robust circuit. The circuit performance is deemed acceptable only if V_{ref} is within 10% of 0.6V, $\tau_s \leq$ 200ns and $V_{do} \leq$ 0.9V. The yield integral can be written in form (1), similar to Section 7.1. 10 MC runs of 10,000 pseudo-random points each were run. 1 QMC run of 10,000 Sobol' points and 9 QMC runs of 10,000 scrambled Sobol' points each, were also run.

7.4 Analysis of Results

Figs. 7 and 9 present the results for all three testcases. Fig. 7 plots the values of the estimates with increasing number of points for each MC (pseudo-random) and QMC (Sobol') run. For all three cases, we can clearly see that the QMC graphs converge more quickly than the MC graphs. In particular, the *non-scrambled Sobol'* points converge *very* fast towards 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 changes the way the space is filled up, and hence, changes the patterns and the discrepancies of the projections of the sequence. What we observe is that changing the patterns in this way causes the QMC performance to degrade in general. This implies that the rank correlation arranges the variables in a way that is optimal (or close to optimal), given the patterns of the non-scrambled LDS. This behavior is more pronounced as the problem dimensionality increases from MSFF to the SRAM Column, suggesting that for low dimensionality (e.g. 31-D MSFF), the LDS is uniformly distributed even for few samples; that is, has few patterns. For high dimensional problems, however, effective variable-dimension mapping should give notable improvement, over a random or uneducated assigment of variables to LDS dimensions. The best estimates shown are computed using all the points from all the MC and QMC runs. These estimates will be used as the "exact" values of the quantities being measured.

Fig. 8 plots the absolute values of the relative QMC estimate errors with two different variable orderings: 1) Correct -- use the rank correlation method outlined in Section 5, and 2) Reverse -- use the reverse ordering, with increasing rank correlation as the dimension increases. For the MSFF we do not see much difference in performance, since the dimensionality is low enough for all dimensions of the LDS to be similarly uniform. For the 122-D bandgap prob-

lem, we can see the that the reversed mapping is slower to converge. For the 403-D SRAM column, the reverse mapping has a larger error after about 700 points. The lower error in the beginning is probably because of good point placement due to chance.

Fig. 9 compares the standard deviation (σ) of the MC runs and the QMC runs with increasing number of points, showing the effectiveness of QMC as a variance-reduction method. The plots are in log-log scale; hence, a $\sigma \propto n^{-\alpha}$ relationship will appear as a straight line with slope $-\alpha$, where n is the number of samples. The plots also show straight lines, fit via least squares, to the standard deviation data, along with the relationship they represent. We can immediately see that QMC exhibits lower variance *and* faster convergence (larger α magnitude) compared to MC in all three cases. Ideally, the value of α should be 0.5 for MC (eqn. (4)). We see that in reality it is a little slower. Even if the convergence rate were to reach this theoretical limit for large n, QMC methods still exhibit faster convergence for the testcases studied. Furthermore,

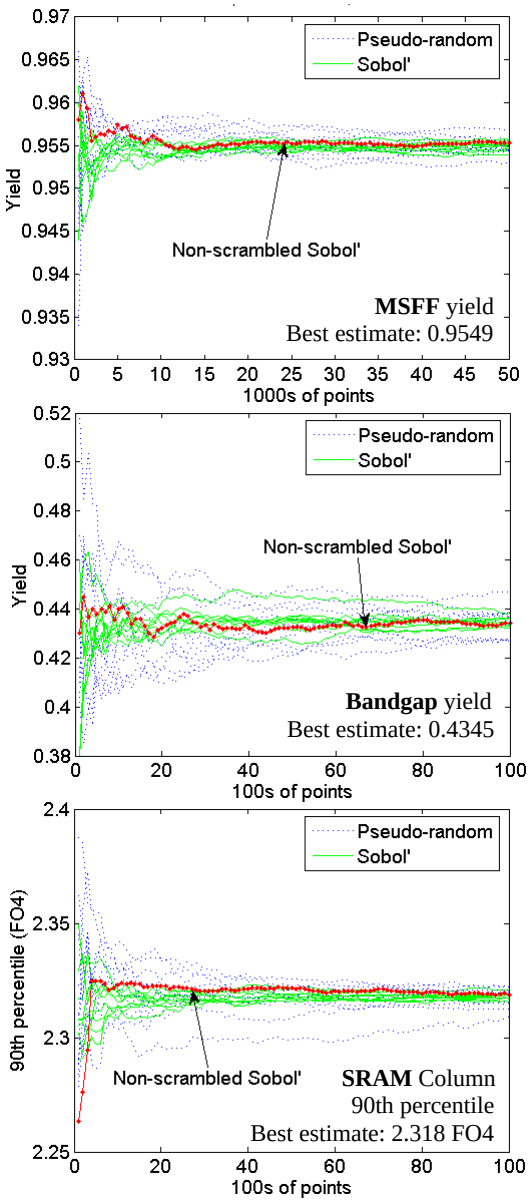


FIGURE 7. Comparison of estimates from Monte Carlo (Pseudo-random) and Quasi-Monte Carlo (Scrambled Sobol').

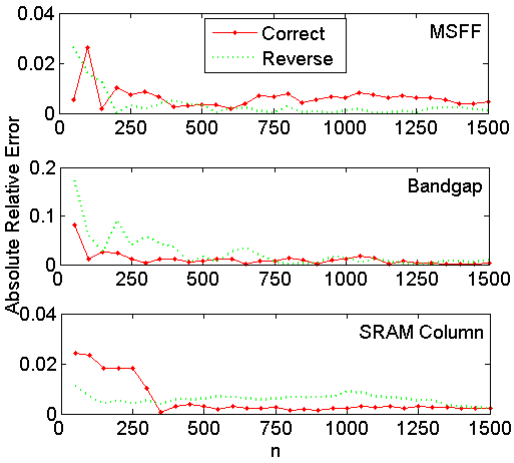


FIGURE 8. QMC estimate with increasing number of points, for the correct variable-dimension mapping (variables sorted with decreasing rank correlation), and for the reversed mapping (increasing rank correlation).

with increasing n, α for QMC should tend towards 1.0 (Section 3), further improving the rate of convergence.

Using these fits, we can estimate the number of MC or QMC samples needed such that the result lies within a given interval for a given confidence level. Using the Central Limit Theorem [30], for a confidence level of 95.45%, this interval is $[\mu - 2\sigma, \mu + 2\sigma]$. Hence, if we want the estimates to lie within 1% deviation from the exact value, with a confidence of 95.45%, the value of σ should be no greater than 0.5% of the exact value. Table 1 compares the number of points needed for MC and QMC, for maximum errors of 1% and 0.1%, at the same confidence level. The exact value is approximated by the best estimate, shown in Fig. 7.

	MSFF		SRAM Col.		Banba bandgap	
95.45% conf. int.	MC/ QMC	speed-up	MC/ QMC	speed-up	MC/ QMC	speed-up
±1%	1114/ 588	**1.9x**	1631/ 354	**4.6x**	89115/ 10360	**8.6x**
±0.1%	180232 /24465	**7.4x**	586771/ 11451	**51.2x**	15182252/ 838062	**18.1x**

TABLE 1. Number of points needed to achieve a given confidence level for given percentage error values.

We can see moderate to large speedups (2x to 50x), showing the effectiveness of QMC as a variance reduction method. These speedups improve as the required accuracy increases. Here, we have assumed that the value of σ computed using 10 runs is exact. This is not true in reality, but, since we are using the same assumption for the MC and QMC cases, the *relative* trends seen here can be believed. It should also be possible to apply other MC variance reduction techniques [2], independently, on top of QMC, to further improve accuracy.

8. Conclusions

Computational finance problems share a number of the features with statistical circuit analysis problems. We demonstrated that one of the most celebrated techniques in the finance world, Quasi-Monte Carlo analysis, can be successfully applied to statistical circuit yield problems, with attractive runtime speedups. However, we also showed that one must be quite careful in mapping these problems onto a QMC form, using appropriate sensitivity information. To the best of our knowledge, this is the largest and most rigorous experimental

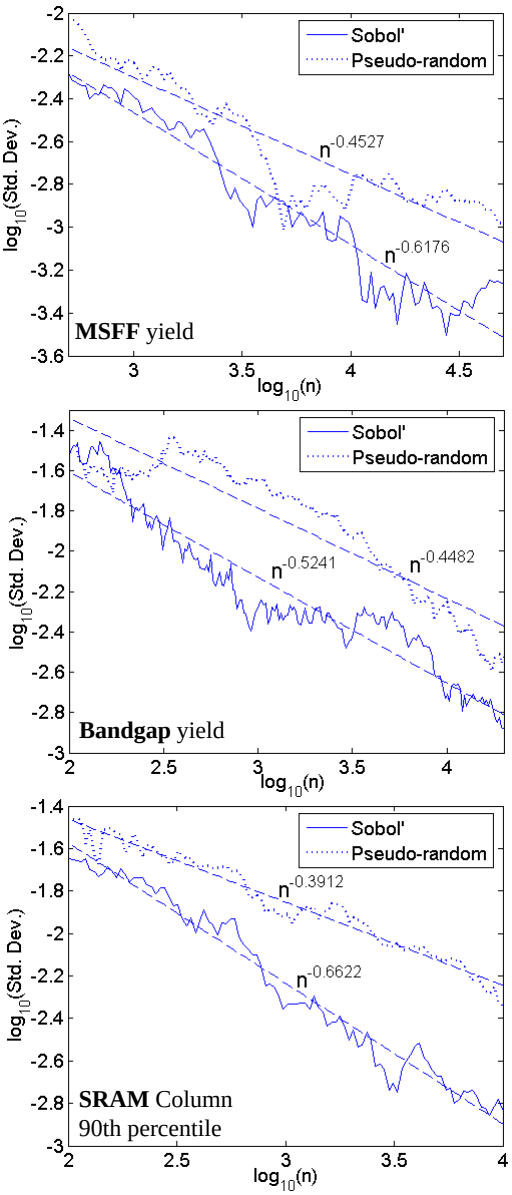


FIGURE 9. Comparison of standard deviation of estimates from Monte Carlo (Pseudo-random) and Quasi-Monte Carlo (Scrambled Sobol').

comparison of MC versus QMC ideas ever undertaken in the context of industrially relevant scaled CMOS technologies and circuits.

Acknowledgements: This work was supported by the MARCO/DARPA Focus Center for Circuit & System Solutions (C2S2), and the SRC.

References

[1] M. Mani, A. Devgan, M. Orshansky, "An Efficient Algorithm for Statistical Minimization of Total Power under Timing Yield Constraints", *IEEE/ACM DAC*, 2005.

[2] P. Glasserman, "Monte Carlo Methods in Financial Engineering", Springer, 2004.

[3] S. Ninomiya, S. Tezuka, "Toward Real-time Pricing of Complex Financial Derivatives", *App. Math. Fin.*, 3(1), pp. 1-20, 1996.

[4] J.H. Halton, "On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-dimensional Integrals", *Nuremische Mathematik*, 2, pp. 84-90, 1960.

[5] A. Papageorgiou, J.F. Traub, "Beating Monte Carlo", *Risk*, 1996.

[6] H. Niederreiter, "Random Number Generation and Quasi-Monte Carlo Methods", *SIAM*, 1992.

[7] E. Hlawka, "Funktionen von beschränkter Variation in der Theorie der Gleichverteilung", *Annali di Matematica Pura ed Applicata*, 54, pp 325-333, 1961.

[8] R.E. Caflisch, W. Morokoff, A. Owen, "Valuation of Mortgage Backed Securities Using Brownian Bridges to Reduce Effective Dimension, *J. Comp. Fin.*, 1, pp.27-46, 1997.

[9] K.-L. Chung, "An Estimate Concerning the Kolmogoroff Limit Distribution", *Trans. Amer. Math. Soc.*, Vol 67, pp. 36-50, 1949.

[10] I.M. Sobol', "The Distribution of Points in a Cube and the Approximate Evaluation of Integrals", *USSR Comp. Math and Math. Phys.*, 7(4), pp. 86-112, 1967.

[11] H. Faure, "Discrépance de Suites Associées à un Système de Numération (en Dimension s)", *Acta Arith.*, 41, pp. 337-351, 1982.

[12] H. Niederreiter, "Low-Discrepancy and Low-Dispersion Sequences", *J. Number Theory*, 30, pp. 51-70, 1988.

[13] H. Niederreiter, C. Xing, "The Algebraic Geometric Approach to Low-Discrepancy Sequences", *Monte Carlo and Quasi-Monte Carlo Methods*, pp. 139-160, 1996.

[14] P. Acworth, M. Broadie, P. Glasserman, "A Comparison of some Monte Carlo and Quasi-Monte Carlo Techniques for Option Pricing", *Monte Carlo and Quasi-Monte Carlo Methods*, pp. 1-18, 1996.

[15] P. Bratley, B.L. Fox, "Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator", *ACM Trans. Math. Soft.*, 14(1), pp. 88-100, 1988.

[16] S. Joe, F.Y. Kuo, "Remark on Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator", *ACM Trans. Math. Soft.*, 29(1), pp. 49-57, 2003.

[17] W.W. Peterson, E.J. Weldon, "Error-Correcting Codes", 2nd ed., MIT Press, 1972.

[18] F.J. Hickernell, "A Generalized Discrepancy and Quadrature Error Bound", *Math. of Comput.*, 67, pp. 299-322, 1998.

[19] X. Wang, K.-T. Fang, "The Effective Dimension and Quasi-Monte Carlo Integration", *J. Complexity*, 19, pp. 101-124, 2003.

[20] G.E. Noether, "Introduction to Statistics: The Nonparametric Way", Springer, 1990.

[21] A.B. Owen, "Randomly permuted (*t,m,s*)-nets and (*t,s*)-sequences", *Monte Carlo and Quasi-Monte Carlo Methods*, pp. 299-317, 1995.

[22] A.B. Owen, "Variance with Alternative Scramblings", *ACM Trans. Model. and Comp. Sim.*, 13(4), pp. 363-378, 2003.

[23] H.S. Hong, F.J. Hickernell, "Algorithm 823: Implementing Scrambled Digital Sequences", *ACM Trans. Math. Soft.*, 29(2), pp. 95-109, 2003.

[24] G. Ökten, W. Eastman, "Randomized Quasi-Monte Carlo Methods in Pricing Securities", *J. Eco. Dyn. and Contr.*, 28(12), pp. 2399-2426, 2004.

[25] M. Matsumoto, Y. Kurita, "Twisted GFSR Generators II", *ACM Trans. Model. and Comp. Sim.*, 4, pp. 254-266, 1994.

[26] S. Tezuka, "Uniform Random Numbers: Theory and Practice", Kluwer, 1995.

[27] W. Zhao, Y. Cao, 'New Generation of Predictive Technology Model for sub-45 Design Exploration", *ISQED*, 2006.

[28] H. Banba et al., "A CMOS Bandgap Reference Circuit with Sub-1-V Operation", *IEEE J. Solid State Cir.*, 34(5), pp. 670-674, 1999.

[29] D. Johns, K. Martin, "Analog Integrated Circuit Design", Wiley, 1996.

[30] R.V. Hogg, A.T. Craig, "Introduction to Mathematical Statistics", 3rd ed., MacMillan, 1971.