Criticality Aware Latin Hypercube Sampling for Efficient Statistical Timing Analysis

Vineeth Veetil, Dennis Sylvester, David Blaauw

EECS Department, University of Michigan, Ann Arbor, MI - 48109

tvvin,dennis,blaauw@eecs.umich.edu

ABSTRACT

Process variation is a major concern in the semiconductor industry today. Probabilistic statistical static timing analysis (SSTA), where random variables are used to represent arrival times, has been proposed as a method to address this challenge. However, there are a number of modeling and accuracy difficulties associated with probabilistic SSTA analysis and optimization methods, such as how to address the skew of arrival times efficiently and combined modeling of drivers and interconnect. In this paper we describe a method to improve the practicality of statistical static timing analysis (SSTA) by focusing on improving the efficiency of Monte Carlo based statistical timing analysis. We introduce a Criticality Aware Latin Hypercube Sampling (CALHS) approach to stratify the process variation space based on critical paths in the circuit and then intelligently sample. The result is that many fewer samples (up to 6.9X on the benchmark circuits studied) are needed to arrive at comparable accuracy in timing estimation compared to a random sampling approach. Also, in comparing a Monte Carlo-based SSTA to traditional SSTA approaches, we find over 50% less error in higher percentile delays for the largest circuits considered, using CALHS, even with a moderate number of samples.

1. INTRODUCTION

Process parameter variations have taken on increasing importance in nanometer-scale CMOS. Rather than using simple corner models that capture worst-case behavior at the device level (and lead to large guardbands), CAD tools today are moving towards a more probabilistic view of circuit timing behavior.

In replacing corner models, there are two primary approaches to incorporating process parameter uncertainty in timing analysis. The first is to perform statistical static timing analysis (SSTA) by modeling gate delay as a function of process parameters and propagating these distribution functions. We refer to these approaches such as those in [5,11] as *traditional SSTA*. In traditional SSTA it has proven challenging to model skewness of the arrival time which results from non-linearity of the gate delays and the maximum function without loss in performance. Also, a number of modeling issues are still in early stages of development, such as combined analysis of large interconnect structures driven by non-linear drivers andcoupling events and modeling of transparent latches. While strong steps have been made to address these issues [5,11,14], it is expected that a mature tradition SSTA tool, capable of performing timing sign-off, may not be widely available for a number of years.

The second approach is to run STA in a Monte Carlo fashion. The importance of Monte Carlo in general has been discussed in [12]. Here, the author proposes that as practical problems involve several dimensions Monte Carlo can form a suitable approach. Such an approach to SSTA would involve selecting samples of the process variation space to obtain statistical distributions of circuit timing behavior (this can be referred to as *Monte Carlo-based SSTA*). Given an accurate underlying process variation model, such techniques are inherently accurate as they do not involve any approximations. The runtime in this approach directly depends on the number of samples employed and has been cited as the major drawback of this approach. The difficulty is that a fully random choice of samples can lead to either a loss of efficiency (too many samples) or accuracy (few non-representative samples). Therefore in Monte Carlo-based SSTA there is a need for variance reduction techniques.

In the past, several variance reduction techniques for parametric yield estimation have been analyzed [1][3][4]. In [1], a Latin Hypercube Sampling Monte Carlo for parametric yield estimation is proposed. However, few results are presented and it is not clear how well the approach will apply to timing analysis. The work in [3] proposes *mixture importance sampling* for statistical SRAM design and analysis. The paper shows the potential of significantly improving the efficiency of circuit analysis using variance reduction techniques. The approach in [4] is to use the *control variates* technique in conjunction with *importance sampling,* for timing yield estimation. However, while several interesting approaches are reviewed, no results are presented.

In this paper, we introduce Criticality Aware Latin Hypercube Sampling (CALHS) for use in Monte Carlo-based SSTA. This contribution is important in the following respects. First, there has been very little work focusing on accurate and efficient timing analysis using Monte Carlo as an alternative to traditional SSTA. Our work is the first to directly study variance reduction aimed at improving the efficiency of Monte Carlo-based SSTA. Second, unlike most previous research on Monte-Carlo based SSTA, our work considers intra-die variation with spatial correlation, based on the model detailed in [5].

This work describes two approaches. First, we use Latin Hypercube Sampling (LHS), which is a known technique in sampling theory, to select samples in the process variation space. STA is performed on these samples, and the distribution of circuit arrival time is obtained. In the second approach, we use timing criticality information to partition the process space into strata. We then use LHS to determine an appropriate set of samples in these strata. As before, we then perform STA on these samples and obtain distributions of circuit timing behavior. We compare these results with a random sampling approach for selecting samples in the process variation space. Only gate length variation is considered in our approach, since it is the most dominant process variation parameter. However, the approach can be easily extended to include additional process variation parameters.

Our experiments show that the CALHS and simple LHS exhibit large speedups relative to a random sampling approach. CALHS is up to 6.9X faster on ISCAS85 benchmark circuits while LHS alone is 2 to 3X faster than random sampling. Also, with a moderate number of samples, over 50% less error is obtained in high percentile delays compared to traditional SSTA.

This paper is organized as follows. Section 2 surveys several important variance reduction techniques in the literature. Section 3 presents our work on variance reduction for Monte Carlo based SSTA. We go on to present results in Section 4 and conclude with Section 5.

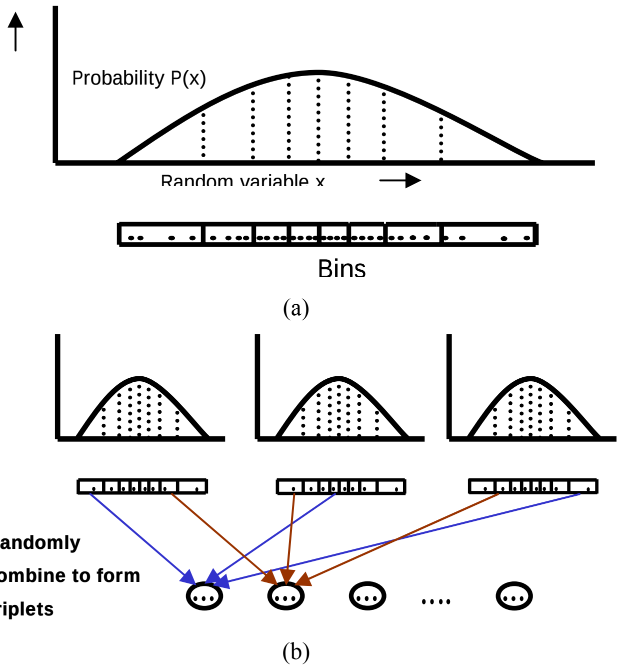
2. Latin Hypercube Sampling

There has been substantial work on mathematical techniques for Monte Carlo variance reduction [6]. Most variance reduction approaches use additional information about the problem at hand to reduce variance. Here, we first look at a few approaches as applied to the yield estimation problem and go on to justify the use of LHS and stratified sampling as the basis for our approach in the timing analysis context.

Importance sampling and *control variates* are techniques that have been studied for application to integrated circuit yield estimation. The yield estimation problem aims at estimating the integral of a binary function over the process variation space. In the *control variates* technique, as applied to the problem of estimating the integral of a function *f(X)*, the idea is to come up with a correlated function *h(X)* whose integral is computed with much less effort. The difference of these functions *f(X) - h(X)* has lower variance than *f(X)* and requires fewer samples. In *importance sampling*, the sampling probability distribution function is chosen to sample more in regions where *f(X)* exhibits higher variation. We refer the reader to [6] for further details on this technique. Another approach in yield estimation is to use the *control variates* technique in conjunction with *importance sampling,* as outlined in [4].

Turning now to the problem of Monte Carlo-based SSTA, it is not immediately clear that the *control variates* technique or *importance sampling* approach of [4] can be effectively applied, as no results are provided. In general, the problem with the *control variates* approach is that it relies on the use of a correlated function *h(X)*. Such a general function is often difficult to find, especially one that considers multiple sources of variation, incorporates various circuit elements like latches, and considers multi-cycle paths. More work is required to establish the effectiveness of these approaches for use in the modern integrated circuit design process. The use of *importance sampling* [4] in yield estimation is justified since it can be argued that there are significantly large regions in process variation space where the circuit is either known to meet the target delay T_C, or at least behave similarly to a model with respect to meeting the target delay. This is not directly applicable to timing analysis however. To summarize, the above methods require further study regarding their applicability in timing analysis.

LHS, on the other hand, does not require any knowledge of the system under consideration, and is therefore general and scalable. LHS attempts to ensure that the samples chosen are spread more or less uniformly in the sample space, across input variables. In other words, its main feature is that it simultaneously stratifies on all input dimensions [9]. The importance of this cannot be overemphasized in a sample space of high dimensionality such as the process variation space. In such a scenario, if a small number of samples are randomly picked, it may severely bias the estimator, and this effect worsens with more dimensions. In a simple version, LHS generates *N* samples from a sample space of *k* variables $X = [X_1, X_2 \ldots X_k]$ in the following way. The range of each variable is partitioned into *N* non-overlapping intervals of equal probability size *1/N*. One value is chosen at random from each of these *N* intervals for every variable. The N values thus obtained for X_1 are randomly paired with the N values obtained for X_2. This gives us N pairs. These are combined randomly with the N values of X_3 to form N triplets. And so on until N k-tuples are obtained. Fig 1 below illustrates LHS for the 3-variable case.



(a)

(b)

Figure 1. LHS sampling with N=8, k=3. (a) Sampling of a variable in equal probability bins. (b) Forming triplets by randomly combining individual samples.

LHS ensures variance reduction in very general cases and can be combined effectively with other techniques for variance reduction. The author of [8] finds that as long as *N*, the number of simulations, is large compared to the number of variables *k,* LHS gives an estimator with lower variance than simple random sampling (referred to as RS for the remainder of the paper) for any function *h(X)* having a finite second moment. Reference [7] cites uses of LHS with *importance sampling* and *control variates.* The author also shows that LHS with *control variates h*as smaller variance than *i.i.d. sampling* (equivalent to RS) with the same *control variates.*

A stratified sampling approach involves partitioning the sample space into mutually exclusive strata, and sampling within individual strata. Fig 2 below illustrates a particular stratification of a 3 variable space into 16 strata.

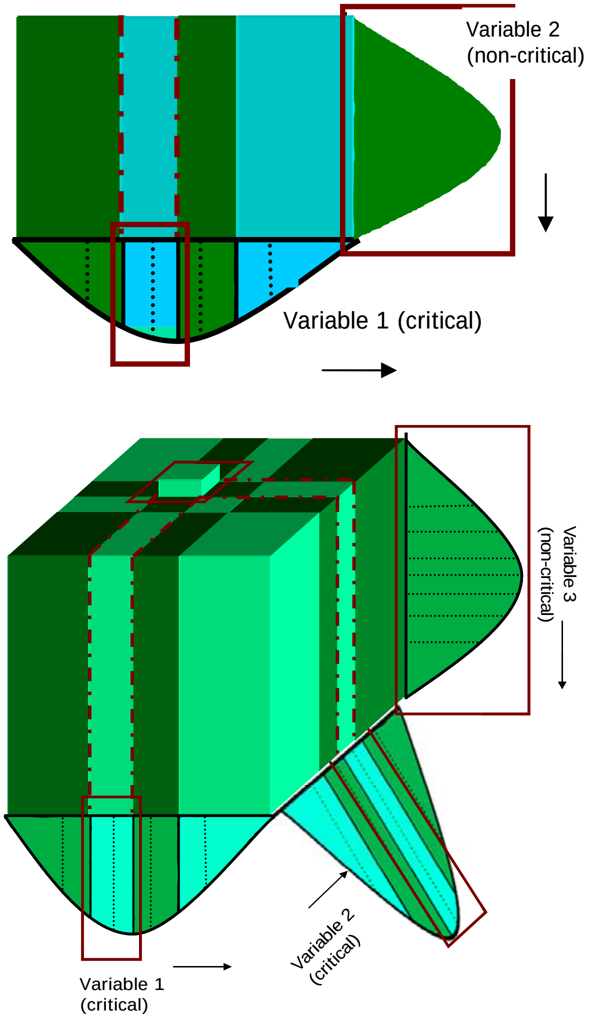


Fig 2. Illustration of stratified sampling in 3-D space. Combinations of 4 regions each in x and y define the strata in this case. A particular stratum and its projections on the 3 components is marked.

In the context of timing analysis, we incorporate the spatial correlation model proposed in [5], where principal component analysis is used to transform the spatially correlated random variables into orthogonal random variables. We perform our sampling on the space of these orthogonal random variables. Another approach would be to directly generate correlated random variables. In [10], the author proposes methods to generate correlated random variables. However, we do not follow that approach in this work for reasons of simplicity and extensibility to other models for manufacturing processes. Thus, our only assumption is that process variation can be represented as a linear combination of orthogonal random variables.

3. Criticality Aware LHS (CALHS) for Timing Analysis

In this section, we propose Criticality-Aware Latin Hypercube Sampling (CALHS) for Monte Carlo-based SSTA.

Statistical STA is defined in [5] as the problem of finding the probability distribution of the max of the path delay distributions for all paths from source node to sink node in the timing graph of the circuit. We concentrate on finding the mean and standard deviation of this distribution, i.e., mean arrival time and standard deviation in arrival time (henceforth μ_{AT} and σ_{AT}, respectively). In Monte Carlo-based SSTA, this is achieved through sampling in the process variation space and finding the average and standard deviation of the worst-case arrival times obtained (μ_{AT} and σ_{AT}) across the multiple instantiations of the circuit. Another parameter is the 99th percentile point in worst-case arrival time, which we refer to as $T_{0.99}$.

As mentioned, our process variation model is based on [5]. We partition the die into *n* x *n* grids. Perfect correlation is assumed for devices within the grid. A single variable represents the variation in each grid, leading to $t = n^2$ correlated variables.

Let $l_g^{\,i} \in \vec{L}_g$ represent the variation of transistor gate length of the i^{th} grid, $\vec{L}_g{}'$ represent the set of principal components computed as in [5]. $l_g^{\,i}$ can be expressed as a linear function of the principal components and the uncorrelated random component:

$$l_g^{\,i} = \mu_{l_g^i} + (a_{i1} \times l_g'^{\,1} + \ldots + a_{it} \times l_g'^{\,t}) + b \times R \qquad (1)$$

where $\mu_{l_g^i}$ is the mean of $l_g^{\,i}$, $l_g'^{\,i}$ is a principal component in $\vec{L}_g{}'$, all $l_g'^{\,i}$ are independent with zero mean and unit variance ; R is the uncorrelated random component. Note that in our case, we have a separate inter-die component with zero mean and unit variance to account for the inter-die variation, apart from the n^2 principal components.

The gate delay, with a Taylor series expansion of first order, is thus a linear combination of principal components of all parameters and the uncorrelated random component [5].

$$d = d_0 + (k_1 \times p_g'^{\,1} + \ldots + k_m \times p_g'^{\,m}) + \eta_d \times R \qquad (2)$$

We have considered only gate length variation as a source of process variation in this work, as discussed before.

In our first approach, we use LHS to sample the process variation space, without using any criticality information. As is clear above, we have *t* i.i.d. principal components with a normal distribution (zero mean and unit variance), a separate inter-die component and N_{gate} uncorrelated random variables. We only consider the principal components and inter-die component (henceforth together referred to as components) for LHS and CALHS – for the uncorrelated random components, we perform random sampling. We first divide the space (-∞, ∞) of each component into 20 bins of equal probability. The choice of 20 is a tradeoff between accuracy (increases with more bins) and number of samples.

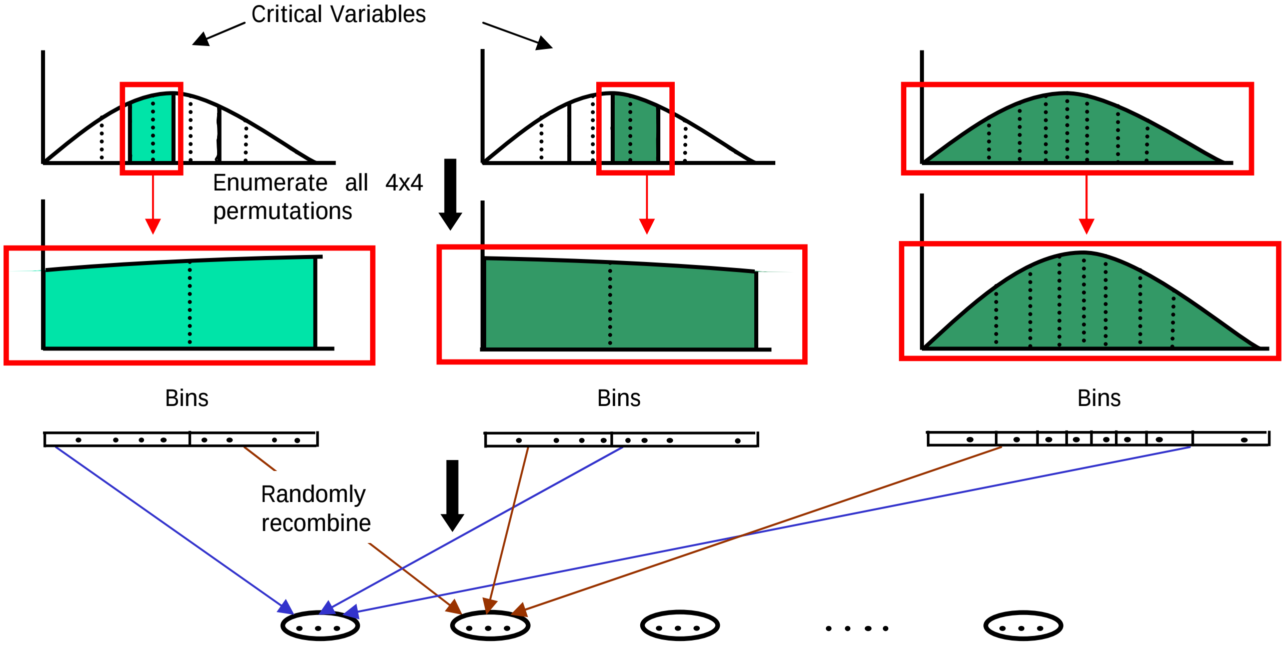


Fig 3. A visual example of Criticality Aware Latin Hypercube Sampling. r.v.$_1$ and r.v.$_2$ are the two critical components in this example.

Now, we perform LHS with N=20, k=t+1 (N=8, k=3 is illustrated in Fig 1). This is repeated *k/20* times to generate *k* samples.

We now extend this approach to include timing criticality information. We find the top *p* components that contribute most significantly to the potential critical paths. For this, we first perform static timing analysis on the nominal circuit to identify critical paths within a slack *s.* Henceforth we will refer to this as the 'criticality slack parameter'. Now, each grid is assigned a weight equal to the number of gates falling in any of the potential critical paths. Let $w_{g,i}$ be the weight of the i^{th} grid. The weight of the j^{th} principal component is given by

$$w_j = \sum_{i=1}^{t} w_{g,i} \times \sigma_i \times corr\,(i,j) \qquad (3)$$

where corr*(i,j)* is the coefficient of the j^{th} principal component in the i^{th} grid variation, and σ_i is the standard deviation of $l_g^{\,i}$. The weight of the inter-die component is given by a similar expression.

The top *p* components (critical components) with highest weights are found in this manner. Now, the idea is to partition the sample space of t+1 components into strata biased towards the critical components, and perform LHS sampling in each such stratum. We will refer to the components in decreasing order of the weight given by (3), i.e., r.v.$_1$ has the highest weight and r.v.$_t$ has the least. Let r.v.$_k$, such that k ≤p, be partitioned into r_k regions of equal probability (similar to the idea of bins mentioned above). Let $R_{k,j} \in R_k$ denote the j^{th} region of *k*, R$_k$ denotes the space of the component *k*. The strata are then defined by:

$$\mathrm{R}_{1,a_1}\mathrm{R}_{2,a_2}\ldots\mathrm{R}_{p,a_p}\mathrm{R}_{p+1}\ldots\mathrm{R}_t$$

$$a_k \le r_k, k \le p$$

We thus have $\prod_{k=1}^{p} r_k$ strata. Fig 2 illustrates stratification for 3 components r.v.$_1$ = x, r.v.$_2$ = y, r.v.$_3$ = z; x and y are the critical components (p = 2), and are divided into 4 regions each, total 16 strata.

Within each stratum, we perform LHS. Consider the highlighted stratum and its projections on the three components in Fig 2. The stratum is defined by the second region of r.v.$_1$ and the third region of r.v.$_2$. The LHS approach for this particular stratum is illustrated in Fig 3. For r.v.$_1$ and r.v.$_2$, we have 2 bins each within the highlighted region, this gives sufficient granularity; for r.v.$_3$, we have 8 bins. This means that we will require 4 samples in each bin of r.v.$_1$ and r.v.$_2$ and just 1 in each bin of r.v.$_3$ to form 8 samples (triplets).

In the timing analysis context, we use 20 bins instead of 8 in the above example. If a critical random variable has 4 regions in the stratification, each region has 5 bins.

4. Experiments and Results

In this section, we present simulation results demonstrating that CALHS leads to a significant speedup in Monte Carlo-based SSTA.

Our simulations are based on a 130nm industrial technology library. The inter-die and intra-die spatial and correlated components of variation are set for an overall standard deviation of 10%. The grid sizes in the spatial correlation model for individual circuits are based on their sizes and vary from smallest 2 by 2 to largest 10 by 10. We perform statistical timing analysis using the three approaches (Random Sampling or RS, LHS, CALHS) on various ISCAS benchmark circuits. It is important to first define a performance metric to compare the approaches.

We have previously defined μ_{AT} and σ_{AT} in Section 3. Consider a Monte Carlo-based SSTA approach. Let the count of Monte Carlo runs be *x*. For many different trials of *x* runs each, μ_{AT} and σ_{AT} yield two distributions (note that for extremely large *x* these distributions should approach delta functions). Let $\mu\ (x)(\mu_{AT})$ and $\sigma\ (x)(\mu_{AT})$ be the mean and standard deviation of the resulting μ_{AT} distribution. We define error in μ_{AT} as a function of *x*:

$$\varepsilon(x)(\mu_{AT}) = 3\sigma(x)(\mu_{AT}) / \mu\ (x)(\mu_{AT}) \qquad (5)$$

Similarly the error for σ_{AT} is

$$\varepsilon(x)(\sigma_{AT}) = 3\sigma(x)(\sigma_{AT}) / \mu(x)(\sigma_{AT}) \qquad (6)$$

These definitions of error capture the fact that as x → ∞ then $\varepsilon(x)(\mu_{AT}) \rightarrow 0$ and $\varepsilon(x)(\sigma_{AT}) \rightarrow 0$ and hence results have converged. Although technically indirect since the error is not directly compared to a golden timing reference (Monte Carlo STA with very large *x*), this is an acceptable metric since it is clear that a tight distribution of μ_{AT} and σ_{AT} will naturally be centered around the correct values, ensuring good overall accuracy. In fact, this is observed to be the case for the μ_{AT} and σ_{AT} from our experiments. For $T_{0.99}$, we use a similar error metric. However, the error mean is not zero for low x, hence we directly refer to a golden timing reference in this case.

For any given method, the minimum *x* such that both $\varepsilon(x)(\mu_{AT}) < 3\%$ and $\varepsilon(x)(\sigma_{AT}) < 3\%$ gives an idea of how small a sample set can yield good results for the approach being investigated. This metric, referred to as the *optimal count* in Tables I and II, is proportional to the runtime and hence evaluates the efficiency of different methods for Monte Carlo-based SSTA.

Table I compares optimal count for the different methods applied to the benchmark set. CALHS here is for the case of a 'criticality slack parameter' of 10ps. The fourth and fifth columns show the speedup of LHS and CALHS over RS. It can be seen that CALHS is consistently faster by a factor of 3.6 to 7X. LHS alone shows a more modest improvement of 1.2 to 2.4X over random sampling.

Table II compares the optimal count required for good accuracy using different values of the criticality slack parameter. This shows that the scheme is stable w.r.t. to the criticality slack parameter.

Figures 3 and 4 illustrate the typical behavior of the error metrics $\varepsilon(x)(\mu_{AT})$ and $\varepsilon(x)(\sigma_{AT})$ described above, shown for one of the largest circuits studied. LHS and CALHS both perform well for $\varepsilon(x)(\mu_{AT})$, while CALHS shows significant improvement over simple LHS for $\varepsilon(x)(\sigma_{AT})$. In both these cases, there is a high improvement over RS. The limiting factor that dictates the minimal amount of acceptable samples (optimal count) is therefore $\varepsilon(x)(\sigma_{AT})$, which explains why CALHS shows much better performance in Table I compared to LHS.

Table I. Comparison of optimal counts for various sampling approaches for Monte Carlo-based SSTA.

Circuit	RS Count	LHS count	CALHS Count	LHS speedup (X)	CALHS speedup (X)
C432	4720	2480	1200	1.9	3.9
C499	6100	2880	880	2.1	6.9
C880	4640	2320	1040	2.0	4.5
C1908	4320	3600	880	1.2	4.9
C2670	4800	2400	1280	2.0	3.8
C3540	6000	3600	1280	1.7	4.7
C5315	4000	3600	1120	1.1	3.6
C6288	4560	2160	960	2.1	4.8
C7552	5120	2160	960	2.4	5.3

Table II Comparison of optimal counts for CALHS using different values of the criticality slack parameter.

Circuit	CALHS Slck-10ps	CALHS Slck-20ps	CALHS Slck-30ps
C432	1200	1200	1200
C499	880	880	880
C880	1040	1040	1040
C1908	720	880	880
C2670	960	1120	1280
C3540	960	1280	1280
C5315	1120	1120	1120
C6288	960	960	960
C7552	800	880	960

Figure 6 below compares the probability distribution of Worst case arrival times for *traditional SSTA,* CALHS and compares with the golden. The golden here is Monte Carlo run of 65k. The circuit under consideration is ISCAS circuit C6288. It is clear that CALHS captures the distribution more accurately when compared to *Traditional SSTA.* Figure 7 compares the accuracy of CALHS and *Traditional SSTA* for the 99th percentile arrival time $T_{0.99}$. In this case, the mean error is always lower for CALHS and becomes 50% of SSTA error at 160 samples.

5. Conclusions

This paper introduced the concept of CALHS with application to timing analysis of integrated circuits. In particular, with growing process variation and the resulting demise of a corner-based

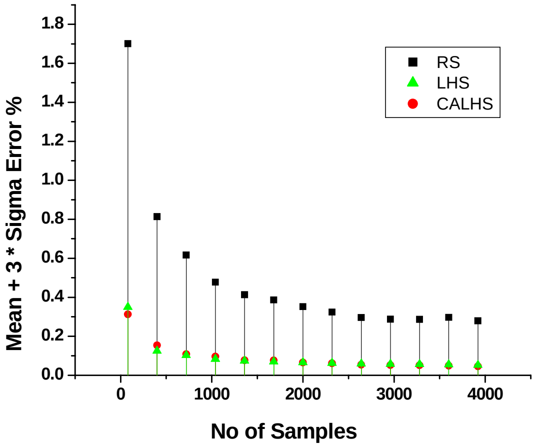


Fig 4. Mean + 3*sigma error in mean arrival time vs number of samples for ISCAS benchmark circuit C6288.

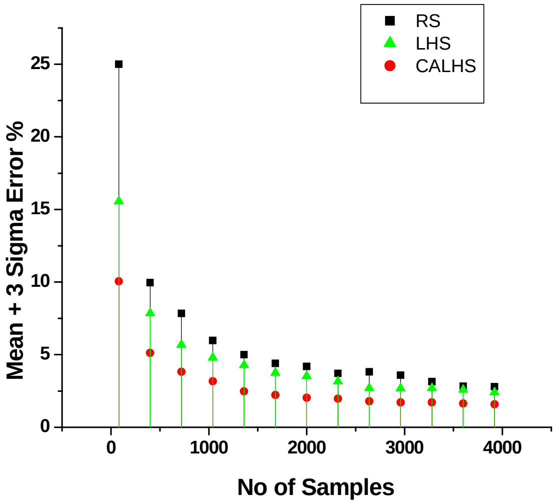


Fig 5. Mean + 3*sigma error in standard deviation of arrival time vs number of samples for ISCAS benchmark circuit C6288.

timing analysis flow, there has been significant interest in statistical static timing analysis, which can result in reduced margins and costs. While most research in this area has centered on so-called traditional SSTA approaches that propagate delay distributions through a timing graph, an alternative approach that is based on Monte Carlo static timing analysis runs also deserves investigation. To make this a viable candidate to replace deterministic STA, intelligent sampling techniques must be applied concurrently to bring down the number of STA runs required to achieve good accuracy.

The CALHS approach in this paper using timing criticality to stratify the process space, and performs LHS within such strata. The result is that many fewer MC runs are needed to generate comparable accuracy in the resulting timing distributions – this leads directly to a much more efficient implementation of MC-based SSTA. Specifically, the criticality awareness provides up to an additional 4.1X reduction (2.7X on average) over LHS in the number of samples needed to achieve 3% error in the timing

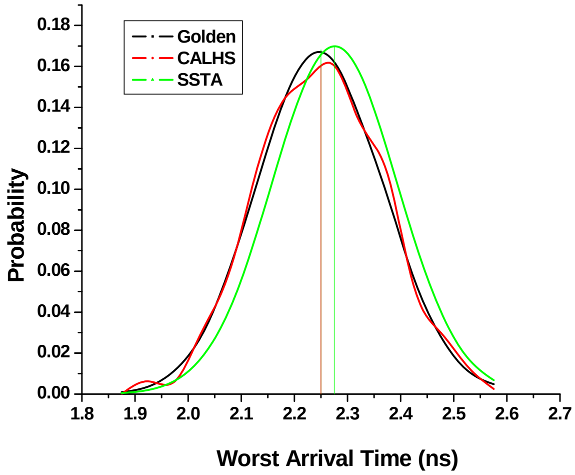


Figure 6. Probability distribution of worst case arrival time for traditional SSTA and CALHS for C6288. These are compared to the golden Monte Carlo count of 65k. Means are highlighted for CALHS and traditional SSTA.

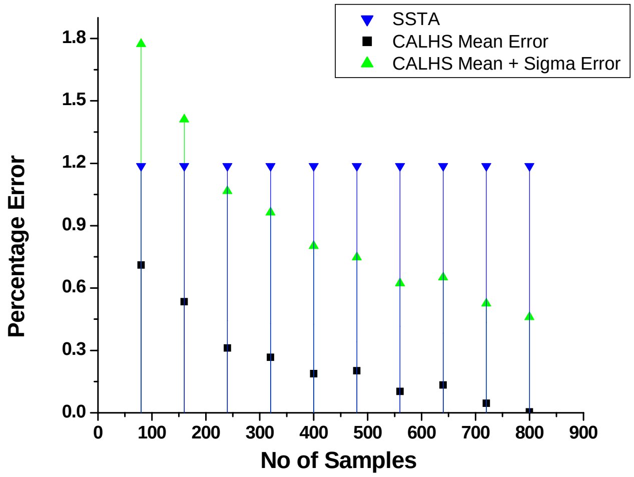


Figure 7. Comparison of error bound for CALHS with traditional SSTA for the 99th percentile arrival time $T_{0.99}$. The circuit considered here is SOVA2.

distribution of the circuit. Overall, CALHS achieves upto 6.9X reduction in number of samples compared to random sampling. Furthermore, we find that MC-based SSTA with CALHS computes the 99th percentile circuit delay with over 50% less error than a traditional SSTA approach for large circuits even at moderate number of samples. These points to both the viability of MC-based SSTA and the need for variation space sampling techniques to further improve this approach. In addition, we note that an efficient Monte Carlo-based SSTA approach can be very valuable in statistical optimization techniques such as described in [13]. In this setting, traditional SSTA has not yet been shown to be scalable [14, 15] as gradient computation is expensive.

Finally, MC-based SSTA is trivially parallelizable, making it an even more interesting approach in practice.

6. REFERENCES

[1] M. Keramat, and R. Kielbasa, "*Worst Case Efficiency of LHSMC Yield Estimator of Electrical Circuits*." *IEEE International Symposium on Circuits and Systems,* v. 3, pp. 1660 – 1663, 1997.

[2] J.A.G. Jess, K. Kalafala, S.R. Naidu, R.H.J.M. Otten, and C. Visweswariah, "*Statistical Timing for Parametric Yield Prediction of Digital Integrated Circuits*," *ACM/IEEE Design Automation Conference,* pp. 932-937, 2003.

[3] R. Kanj, R. Joshi, and S. Nassif, "*Mixture Importance Sampling and Its Application to the Analysis of SRAM Designs in the Presence of Rare Failure Events*," *ACM/IEEE Design Automation Conference* , pp. 69-72, 2006.

[4] S. Tasiran, and A. Demir*, "Smart Monte Carlo for Yield Estimation*," *ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*. (TAU), 2006.

[5] H. Chang, and S.S.Sapatnekar, "*Statistical Timing Analysis Considering Spatial Correlations using a Single Pert-Like Traversal*," *IEEE/ACM International Conference on Computer-Aided Design (ICCAD),* pp. 621-625, 2003.

[6] R.Y. Rubinstein, *Simulation and the Monte Carlo Method,* John Wiley & Sons, Inc., 1981.

[7] A.B. Owen, *A Central Limit Theorem for Latin Hypercube Sampling,* in *Journal of the Royal Statistical Society B,* 54, No.2, pp. 541-551., 1992.

[8] M. Stein, *Large Sample Properties of Simulations Using Latin Hypercube Sampling, Technometrics,* 29, 143-151, 1987.

[9] W.L. Loh, *On Latin Hypercube Sampling, The Annals of Statistics,* Vol. 24, No. 5, 2058-2080, 1996.

[10] A.B. Owen, *Controlling Correlations in Latin Hypercube Samples, Journal of the American Statistical Association,* Vol. 89, No. 428, pp. 1517-1522, Dec 1994.

[11] C. Visweswariah, K. Ravindran, K. Kalafala, S.G.Walker, S. Narayan, "*First-Order Incremental Block-Based Statistical Timing Analysis,*" *ACM/IEEE Design Automation Conference,* pp 331-336, 2004.

[12] L.Scheffer*., "The Count of Monte Carlo*," *ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*. (TAU), 2004.

[13] S.H.Kulkarni, D.Sylvester, D.Blaauw, *"A Statistical Framework for Post-Silicon Tuning Through Body Bias Clustering", International Conference on Computer-Aided Design (ICCAD),* 2006.

[14] K.Chopra, S.Shah, A.Srivastava, D.Blaauw, D.Sylvester, *"Parametric yield maximization using gate sizing based on efficient statisticalk power and delay gradient computation", Proc. International Conference on Computer Aided Design(ICCAD),* pp 1023-1028, 2005.

[15] J.Xiong, V.Zolotov, N.Venkateswaran, C.Visweswariah, *"Criticality Computation in Parametrized Statistical Timing", Proc Design Automation Conference,* pp 63-68, 2006.