Statistical Moment Estimation in Circuit Simulation

Ashish Nigam, Qin Tang, Amir Zjajo, Michel Berkelaar, Nick van der Meijs
Circuit and System Group, Delft University of Technology
Mekelweg 4, 2628 CD, Delft, The Netherlands
A.Nigam@tudelft.nl

***Abstract*—Monte Carlo methods and simulation are often used to estimate the mean, variance, and higher order statistical moments of signal properties like delay and slew. The main issues with Monte Carlo methods are the required long run time and the need for prior detailed knowledge of the distribution of the variations. Additionally, most of available circuit simulation tools can run Monte Carlo analysis for Gaussian, lognormal and uniform distribution only. In this paper, in order to estimate these statistical moments, we propose a new method based on the uniform sampling technique and weighted sample estimator. The proposed method needs significantly less simulation runs, and does not need detailed prior knowledge of the variation distributions. Furthermore, it can be used for any type of probability distribution irrespective of the circuit simulation tool used for the analysis. The results obtained shows that the proposed method needs 100× fewer simulations iterations than Monte Carlo runs for the moments estimation of the delay for standard cells in 45nm and 32nm technologies.**

I. Introduction

The delay of a logic gate strongly depends on variations in process, voltage, and temperature (PVT). As we are moving towards nanometre technology, the process variation is increasing, causing significant uncertainty in the delay estimation [1] and greatly impacting the yield [2], [3]. As a consequence, the accuracy of the conventional static timing analysis (STA) with corner based approach for estimation of digital circuit performance in advance technology processes is a serious concern [4]. Due to these PVT variations, the delay is a statistical parameter instead of a deterministic one. The process of estimating the delay of a data path with PVT variation is known as Statistical STA (SSTA) [5], [6], [7], [8].

In SSTA, the standard cell delay and signal slew are stochastic parameters, and their parameters are often specified with their statistical moments. Practically, Monte Carlo (MC) is the dominant method of choice for statistical moment estimation of these parameters [9], [10]. However, the standard Monte Carlo has the following two limitations.

First, due to the underlying principle of MC analysis, a large number (thousands) of simulation iterations are required for moment estimation with a high confidence bound. Due to the large number of cells in standard cell libraries and long simulation times for advanced transistor models, the necessity of thousands of simulation iterations results into very long circuit simulation run times. Practically, the high run times required for SSTA library characterization, limits its usefulness for large scale circuits.

Second, due to the nature of semiconductor manufacturing processes and circuit behaviours, the PTV parameters typically do not follow a Gaussian distribution [4]. Furthermore, their non-linear relation with delay and slew may result into non-Gaussian distribution of the delay and slew. However, the state of the art circuit simulation tools (e.g. Cadence Spectre [11]) can only run MC with Gaussian, lognormal and uniform distributions, and, unfortunately, forcing any non-Gaussian PVT into these distributions can lead to large errors. To deal with this issue, several non-Gaussian SSTA methodologies have been proposed [12]. These methodologies require higher order moments for accurate modelling of the variations. Additionally, the higher order moments further increase the simulation iterations required in MC iterations.

Several research efforts have been made to speedup the standard Monte Carlo method by improving the random sampling method of the parameters, e.g. Latin Hypercube Sampling (LHS) [13], Quasi Monte Carlo (QMC) [14], and Stratification + Hybrid QMC (SH-QMC) [15]. However, the parameters sampling in the circuit simulations are still their distributions dependent and not applicable for various types of probability density function.

In this paper, we propose a fast statistical moment estimation (FSME) method, which provides two major advantages over standard MC: first, the FSME method can use any probability density function (pdf) irrespective of the simulation tools, and second, for the same accuracy as MC, the FSME method requires two orders of magnitude fewer simulation iterations which results into 100× speedups in the library characterization. The application of the FSME method is not only limited to digital circuit design and SSTA; it is equally applicable in analog circuit design.

II. Fast Statistical Moment Estimation Method

The standard MC method is based on the random sampling of the parameters of interest based on their pdf. This procedure takes more samples around the parameter values with high probability than around the less probable values. Since the sampling method depends on the pdf of the parameters, a large number (thousands) of samples are normally required to generate enough samples for less probable values. Additionally, the dependence on the pdf of the corresponding process parameter makes it necessary to provide the statistical details of the parameter variation before the starting of the simulation. The circuit simulation is repeated for each set of sampled parameters. This results into long run times and high memory requirement to store all the data. The desired simulation output is measured in each simulation, leading to the sample set

of measured values. The moments of the circuit simulation outputs are calculated using standard moment estimation equations on the sample set.

In contrast, by using the FSME method, the probability distribution of the process parameter and the circuit simulation are decoupled. In the proposed method, instead of randomly sampling, the space is sampled with uniform distribution. Moreover, to accurately estimate the statistical moments, a weighted sample estimator is utilized. The process involved in circuit simulation and data processing are discussed below.

A. Circuit Simulation

Unlike MC method, the FSME method runs the simulation with a uniformly spaced parameter sweep, which ensures the required coverage of each simulation parameter, e.g. if a parameter is following a Gaussian distribution then $\pm 3\sigma$ spread around its mean value is sufficient. This implies that the range of the parameter sweep in the simulation needs to be close to the spread of the real parameter distribution. Note that this is the only link required between the real parameter distribution and the data needed to perform simulations.

Let us assume that X are the process parameters (e.g. effective channel lenght L, channel width W, threshhold voltage V_{th}, etc.), where X is a set of vectors X_i, with each vector X_i corresponding to the sampled points $X_i[j_i]$ of the i^{th} parameters, and that Y is a vector of the simulation output $Y[k]$ (e.g. delay, slew, etc.). X and Y will be used in the data processing step to estimate the statistical moments of the output.

B. Data Processing

The statistical moments of the circuit simulation output (Y) depend on the probability of each simulation run, which in turn depends on the probability of each process parameter (X_i) used in the simulation. As a result, the *pdf* of each process parameter is required in the data processing step. In the proposed method, each simulation is considered as an event. The probability of each event is estimated first, followed by the moment estimation of the output.

In the probability space, each simulation is a discrete random event which is associated with a probability based on the value of the process parameters of the particular simulation and its *pdf*. The process parameter X_i can take any value with an infinite number of possibilities within the spread of the process parameter, leading to an almost zero probability in the continuous domain. However, the simulation is carried out only for certain values of the process parameters $X_i[j_i]$, and each sampled value of the process parameter is associated with a certain probability. The probability of each discrete process parameter $X_i[j_i]$ is estimated from the given *pdf* of X_i. Followed by this, the probability of each discrete experiment event k is estimated from the probability of the discrete process parameter values.

1) Probability of Process Parameter: The following notation for the probability and the *pdf* function will be further used in the paper

$$P_d() \rightarrow \text{Probability of discrete variable}$$
$$P_c() \rightarrow \text{Probability of continuous variable}$$
$$P_s() \rightarrow \text{Probability of discrete simulation event}$$
$$f_i() \rightarrow \text{Probability density function of } X_i$$

For a stochastic process parameter X_i, its probability and its *pdf* are related with

$$P_c(X_i[m] < X_i \leq X_i[n]) = \int_{X_i[m]}^{X_i[n]} f_i(x)dx \tag{1}$$

Let us assume that X_i is a uniformly sampled process parameter with a sampling step of ΔX_i, under the constraint that ΔX_i is much smaller than the standard deviation σ_{X_i}. $X_i[j_i]$ are the sampled values of the X_i, which are used in the circuit simulation with uniform sampling. The vector $X_i[j_i]$ can be written as:

$$X_i[j_i] = [\cdots, -2\Delta X_i, -\Delta X_i, 0, \Delta X_i, 2\Delta X_i, \cdots] \tag{2}$$

where

$$\Delta X_i \ll \sigma_{X_i} \tag{3}$$

Let us define the probability of a discrete variable $X_i[j_i]$ to be equal to the probability of a continuous variable X_i varying from $(X_i[j_i] - \Delta X_i/2)$ to $(X_i[j_i] + \Delta X_i/2)$. Since ΔX_i is much smaller than σ_{X_i}, piecewise constant (PWC) approximation can be used to evaluate the integration of the *pdf*

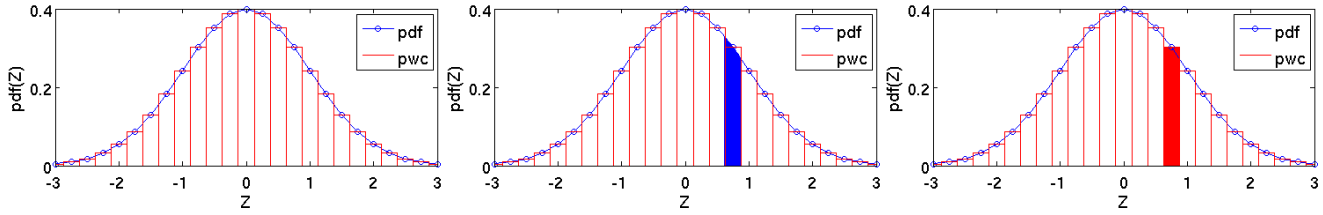
$$\begin{aligned} P_d(X_i[j_i]) &= P_c\left(X_i[j_i] - \frac{\Delta X_i}{2} < X_i \leq X_i[j_i] + \frac{\Delta X_i}{2}\right) \\ &= \int_{X_i[j_i]-\Delta X_i/2}^{X_i[j_i]+\Delta X_i/2} f_i(x)dx \\ &\approx f_i(X_i[j_i]).\Delta X_i \qquad \text{PWC approx.} \\ \Rightarrow P_d(X_i[j_i]) &= f_i(X_i[j_i]).\Delta X_i \qquad \text{if } \Delta X_i \ll \sigma_{X_i} \end{aligned} \tag{4}$$

Thus, the probability of the discrete process parameter $X_i[j_i]$ is equal to the integral of the *pdf* around $X_i[j_i]$ within the bound of $\pm\Delta X_i/2$, and piecewise constant approximation can be used to simplify the integration.

To illustrate it with the example, consider PWC approximation of a Gaussian distributed random variable Z with zero mean and unit variance as shown in Figure 1a. Integration of this *pdf* around some $Z[l]$ and PWC approximation for integration around the same $Z[l]$ are shown with a filled bar in Figures 1b and 1c, respectively. In this approximation, the *pdf* values higher than the *pdf* at $Z[l]$ are decreased to $pdf(Z[l])$, and the *pdf* values lower than the *pdf* at $Z[l]$ are increased to $pdf(Z[l])$. The errors introduced by these changes are having opposite sign and this neutralization effect reduces the error due to the approximation. The total error is reduced by increasing number of samples during circuit simulation.

If X_i is sampled from $-\infty$ to $+\infty$, then the sum of the probability of all discrete values will be equal to one

$$\sum_{\text{all } j_i} P_d(X_i[j_i]) \approx \int_{-\infty}^{\infty} f_i(x)dx = 1 \tag{5}$$



(a) *pdf* and *pwc* (b) *pdf* integration (c) *pwc* integration

Fig. 1: Piecewise constant approximation of probability density function

In our example, if Z follows the Gaussian distribution, then $\pm 3\sigma_Z$ spread of Z around its mean covers 99.8% of the probability space

$$\int_{-3\sigma_Z}^{3\sigma_Z} pdf(z)dz = 0.998 \tag{6}$$

Consequently, if $Z[l]$ is sampled within the range of $\pm 3\sigma_Z$ around its mean, then the sum of the discrete values $Z[l]$ will cover 99.8% of the probability space

$$\sum_{\text{all } l} P_d(Z_l) \approx 0.998 \tag{7}$$

The range of the process parameter sweep in the simulation can be changed based on the requirement of the probability coverage.

2) Probability of Simulation: The probability of each discrete simulation event (k) is equal to the joint probability of all process parameters

$$P_s(k) = P_d(X_1[j_1], X_2[j_2], \dots) \tag{8}$$

In general, the process parameters are not independent. In order to simplify the data processing step, principal component analysis (PCA) can be used to convert the correlated process parameters into uncorrelated simulation parameters, under constraints of the speed-accuracy trade-off. Hence, without loss of generality, the parameters X_i used in this paper are assumed to be independent after PCA. Assuming that the joint probability of independent random variables is equal to the product of the probability of each random variable, the probability of each discrete simulation event can be rewritten as

$$P_s(k) = P_d(X_1[i]).P_d(X_2[i]) \dots \tag{9}$$

Let us define $P_d(Y[k])$ as the probability of the output $Y = Y[k]$ due to the experiment k only. Since the probability of the simulation event k is $P_s(k)$, we can define $P_d(Y[k])$ as follows:

$$P_d(Y[k]) = P_s(k) \tag{10}$$

$$\Rightarrow P_d(Y[k]) = P_d(X_1[i]).P_d(X_2[i]) \dots \tag{11}$$

Each experiment k gives an outcome $Y[k]$. The unknown probability of obtaining this outcome, $P_d(Y[k])$, is estimated from the known joint probability of the process parameters in the sample point k. Because of the assumption of independence, this joint probability is the product of the probabilities of each individual parameter.

Note that the $P_d(Y[k])$ is not the probability of $Y = Y[k]$, as more than one experiments could produce the same value of the output $Y[k]$ in case of non-monotonous function of Y. Each $P_d(Y[k])$ will have different probability value depending on the probability of the experiment k.

3) Moment Estimation: A weighted sample estimator is used here for estimating the moments of the output parameter Y [9]. To illustrate this process, consider the circuit simulation run N times. Let us assume that the probability of each simulation output $Y[k]$, i.e. $P_d(Y[k])$, is already estimated in the previous step. The probability of the each output $Y[k]$ implies that the output event $Y[k]$ should repeat itself by $\lceil N.P_d(Y[k]) \rceil$ times. To obtain a high accuracy, each simulation output $Y[k]$ should occur at least once, implying that the lower bound of N should be defined as

$$N \geq \frac{1}{min(P_d(Y[k]))} \tag{12}$$

Now, let us define a vector $R(Y[k])$ as a set of experiment outputs $Y[k]$ which, repeats $\lceil N.P_d(Y[k]) \rceil$ times, i.e.

$$R(Y[k]) = [Y[k], Y[k], \dots] \qquad \lceil N.P_d(Y[k]) \rceil \text{ times} \tag{13}$$

where the outcome (O) can be written as

$$O = [R(Y[1]), R(Y[2]), R(Y[3]), \dots] \tag{14}$$

Once the outcomes (O) have been generated, the statistical moments of simulation output Y can be evaluated using standard moment estimation equations on the sample set, where the mean (μ), variation (σ^2), and normalized n^{th} central moment (μ_n) are given as

$$\mu_y = E(O) \tag{15}$$

$$\sigma_y^2 = E((O-\mu)^2) \tag{16}$$

$$(\mu_n)_y = E((O-\mu)^n)/\sigma^n \tag{17}$$

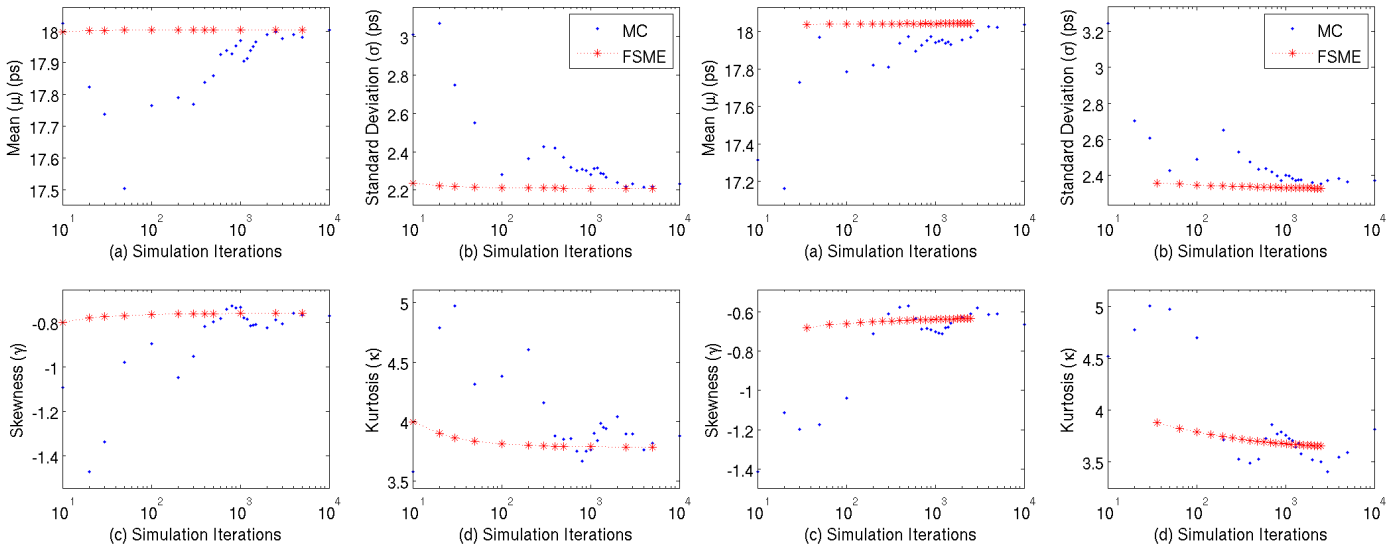
These equations can be rewritten using (13) and (14) as

$$\mu_y = \frac{\sum Y[k].P_d(Y[k])}{\sum P_d(Y[k])} \tag{18}$$

$$\sigma_y^2 = \frac{\sum Y[k]^2.P_d(Y[k])}{\sum P_d(Y[k])} - \left[\frac{\sum Y[k].P_d(Y[k])}{\sum P_d(Y[k])}\right]^2 \tag{19}$$

$$(\mu_n)_y = \sum_{m=0}^{n} \frac{\binom{n}{m}(-\mu_y)^m \sum Y[k]^{n-m}.P_d(Y[k])}{(\sigma_y)^n.\sum P_d(Y[k])} \tag{20}$$

Note that N is only used to develop the outcome O during the illustration of the process of estimating of the moments. When rewriting (15), (16), and (17) using (13) and (14), N



(a) 45nm Inverter - One parameter variation

(b) 45nm Inverter - Two parameter variations

Fig. 2: The first four moment estimation vs simulation runs for MC and FSME with one parameter (L) and two parameters (L and W) variations in 45nm Inverter

appears in both the numerator as well as denominator, and it cancels each other. Hence, N is not required in the final moment estimation equations.

Using the method described above, the statistical moments of various non-Gaussian probability density functions can be estimated irrespective of the simulation tool. The proposed sampling approach of parameter values requires fewer simulations leading to a much faster conversion of the moments. Moreover, since the exact process variation distribution is not required during the simulation run, a slight change in the process variation spread can be analyzed without rerunning the circuit simulation.

III. Simulation Results and Comparison

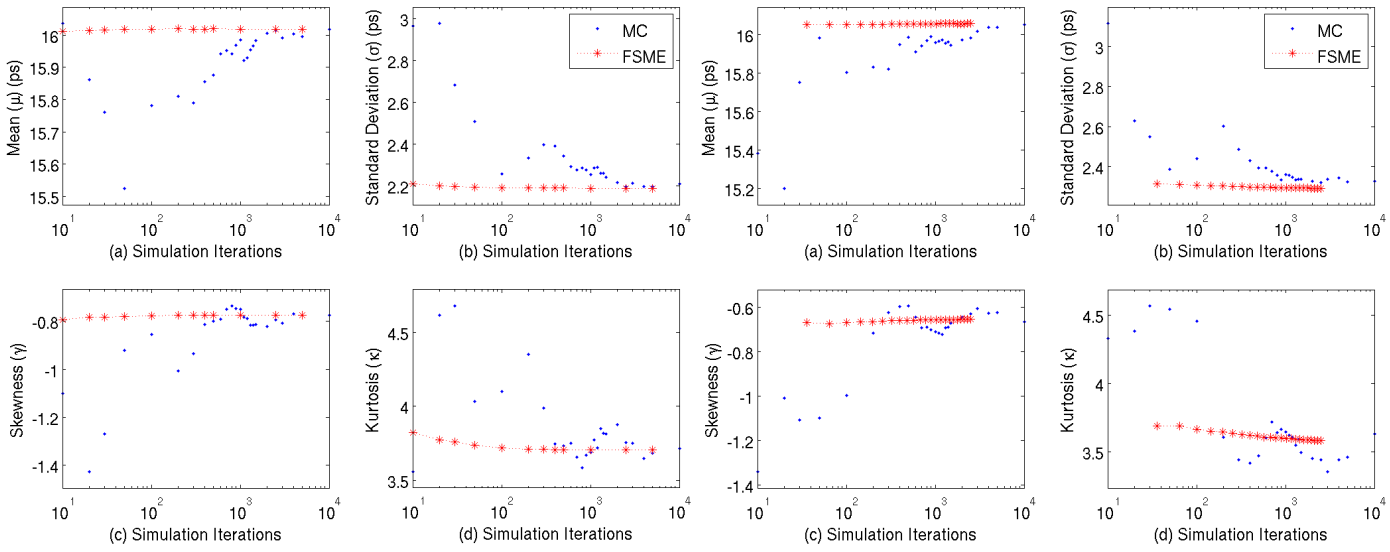
To evaluate the accuracy of the FSME method, extensive Spectre circuit simulations have been carried out with the FSME method as well as with the standard MC method. The results of both simulation methods are reported and compared below.

In the experimental setup, 45nm and 32nm predictive technology models (PTM) have been used for all simulations [16]. Five different circuits (Inverter, Buffer, NAND, NOR, 5 Inverters Chain) have been used and all these standard cells were sized according to their corresponding predictive technology model [17]. The process variations are considered to be a Gaussian distribution such that the results can be compared with the standard MC results. The proposed method is scalable for any number of parameters variations and various process parameters can be used, e.g. L, W, V_{th}, etc. However, due to the space limitation, only two sets of variations are discussed here. In first set, the variation is considered in the effective channel length (L) of the MOSFET with $3\sigma_L$, which equals to 20% of the nominal value of L. In addition to the first set, the variation in the effective channel width (W) is also considered in second set with $3\sigma_W$, which is equivalent to 20% of the nominal value of W. In the output, the first four statistical moments (mean [μ], standard deviation [σ], skewness [γ] and kurtosis [κ]) of the delay of the standard cell have been estimated. Cadence Spectre was used for circuit simulation and Matlab for data processing.

The first four moments (μ, σ, γ, and κ) of the delay vs simulation runs for the 45nm inverter with first set of variation using MC and FSME are shown in Figures 2a. Similarly, these moments of the delay vs simulation runs for the 45nm inverter with second set of variations are shown in Figures 2b. It is clear from these plots that FSME converges much faster than MC. The scattered plot of MC is due to its random sampling nature. As a result of the better convergence of FSME, the best available moments estimates from the FSME are taken as a golden reference value from both sets of variations for FSME and MC run comparison.

The error for the first set of variation after five thousand iterations in MC and fifty iterations in FSME with respect to the respective reference value is reported below. The error in the mean estimation using five thousand iterations of MC with reference value is 0.133% whereas fifty iterations of FSME has only 0.006% of error. Similarly, the error for standard deviation estimation using MC is 1.059% whereas the FSME has an error of only 0.245%. The MC error in skewness estimation is 1.598% and FSME gives an error of only 1.257%. Lastly, kurtosis estimation has an error of 2.489% in MC where as FSME is at 1.304% error margin.

Since two parameter variations needs more simulation iterations, ten thousand iterations in MC and hundred iterations in FSME are used to estimate the error with the respective reference value. The error in the mean estimation using ten thousand iterations of MC with reference value is 0.03% whereas hundrad iterations of FSME has only 0.019% of error. Similarly, the error for standard deviation estimation using MC



(a) 32nm Inverter - One parameter variation

(b) 32nm Inverter - Two parameter variations

Fig. 3: The first four moment estimation vs simulation runs for MC and FSME with one parameter (L) and two parameters (L and W) variations in 32nm Inverter

is 1.772% whereas the FSME has an error of only 0.817%. The MC error in skewness estimation is 4.637% and FSME gives an error of only 4.038%. Lastly, kurtosis estimation has an error of 4.432% in MC where as FSME is at 3.663% error margin.

It is clear from above experimental results that MC with five thousand iterations produce more inaccurate results in comparison to the respective golden reference than FSME with only fifty simulations for one parameter variations. Equally, for two parameter variations, MC with ten thousand iterations has more error in comparison to the respective golden reference than FSME with only hundred simulations for one parameter variations. Similar behaviour is observed in all the five test circuits in both the 45nm and the 32nm technologies. The reference value of these four moments along with the error in the moment estimation for MC with five thousand runs and FSME with fifty runs using 45nm and 32nm technology with first variation set are reported in Table I. Similar table for second set of variations is reported in Table II. The plots of the moment estimation vs simulation runs for Inverter in 32nm PTM using first and second set of variations are shown in Figures 3a and 3b, respectively. Furthermore, Buffer, NAND, NOR, and Inverter Chain have similar behaviour, thus their plots are not included here.

In the results above, we assumed that the variations are following Gaussian distribution only. Now, four different probability density functions (Gaussian, Lognormal, Gamma, and Beta) with the same mean and standard deviation have been considered for the first set of variations. The first four moments of the delay vs simulation runs for 45nm buffer using these probability density functions are shown in Figure 4. In the process of these moment estimations, only the mathematical implementation of the *pdf* function is changed, and rerunning of the simulation is not required. It is clear from the figure

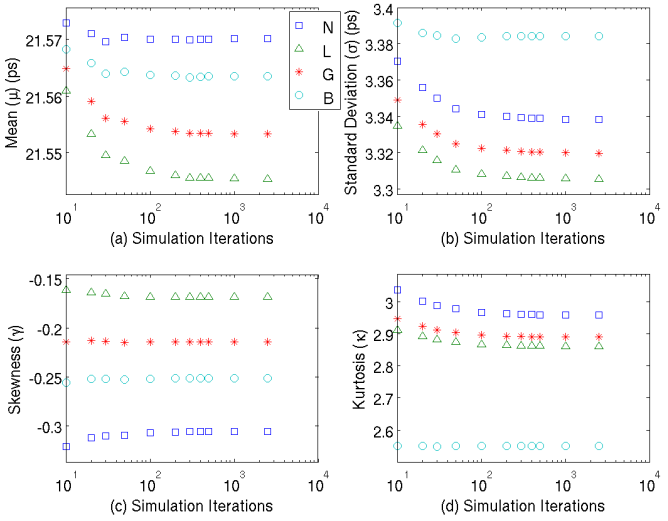


Fig. 4: Moment estimation vs simulation run in 45nm Buffer using Gaussian (N), Lognormal (L), Gamma (G), and Beta (B) distributions.

that the higher order moments significantly vary with the distribution of the parameters.

The above results show that the simulation iterations required in FSME to estimate the moments differ from the simulation iterations required in MC by two orders of magnitude. This results into 100× speedup in the library characterization. Furthermore, different parameter spread can be analyzed in FSME only by changing the parameter of the *pdf* function in the data processing stage. Moreover, any type of probability density function can be used with FSME by changing the implementation of the *pdf* function only. This extra data processing does not require rerunning of the circuit simulator, which results into faster run times and smaller memory requirement to store all the data.

TABLE I: Error % comparison in the first four moments estimation for one parameter (L) variation using Monte Carlo (5000 runs) and proposed method (50 runs) in the 45nm and 32nm PTM technology

	Mean (μ)			Standard Deviation (σ)			Skewness (γ)			Kurtosis (κ)		
Circuits	**Ref (ps)**	**MC %**	**New %**	**Ref (ps)**	**MC %**	**New %**	**Ref**	**MC %**	**New %**	**Ref**	**MC %**	**New %**
45nm PTM Technology												
Inverter	18.004	0.133	0.006	2.210	1.059	0.245	-0.759	1.598	1.257	3.786	2.489	1.304
Buffer	21.570	0.167	0.001	3.338	0.991	0.173	-0.305	1.853	1.359	2.958	2.128	0.674
NAND	25.913	0.103	0.004	2.473	1.093	0.267	-0.682	2.015	1.381	3.685	3.218	1.336
NOR	20.773	0.127	0.020	2.369	1.176	0.304	-0.804	3.659	1.384	3.997	5.659	1.506
Inverter Chain	37.558	0.158	0.001	5.530	1.055	0.207	-0.253	2.176	1.144	2.981	2.622	0.775
32nm PTM Technology												
Inverter	16.019	0.148	0.005	2.187	0.915	0.233	-0.773	0.117	0.811	3.704	0.527	0.824
Buffer	18.082	0.203	0.003	3.409	0.922	0.215	-0.186	3.481	0.743	2.729	2.125	0.452
NAND	23.624	0.115	0.004	2.513	0.999	0.251	-0.607	0.836	1.154	3.580	1.165	1.041
NOR	17.723	0.141	0.006	2.298	1.158	0.310	-0.875	2.817	1.203	4.100	4.490	1.413
Inverter Chain	29.928	0.184	0.005	5.155	0.986	0.189	-0.231	3.324	0.418	2.864	2.097	0.591

TABLE II: Error % comparison in the first four moments estimation for two parameters (L and W) variation using Monte Carlo (10000 runs) and proposed method (100 runs) in the 45nm and 32nm PTM technology

	Mean (μ)			Standard Deviation (σ)			Skewness (γ)			Kurtosis (κ)		
Circuits	**Ref (ps)**	**MC %**	**New %**	**Ref (ps)**	**MC %**	**New %**	**Ref**	**MC %**	**New %**	**Ref**	**MC %**	**New %**
45nm PTM Technology												
Inverter	18.042	0.030	0.019	2.328	1.772	0.817	-0.633	4.637	4.038	3.656	4.432	3.663
Buffer	21.592	0.001	0.010	3.374	1.445	0.692	-0.290	4.593	3.689	2.976	3.492	2.134
NAND	25.965	0.022	0.011	2.653	1.813	0.810	-0.523	5.476	4.875	3.535	5.099	3.831
NOR	20.810	0.031	0.033	2.477	1.982	0.983	-0.699	8.667	5.072	3.880	9.717	4.932
Inverter Chain	37.598	0.006	0.007	5.574	1.500	0.700	-0.247	4.332	3.114	3.001	4.181	2.422
32nm PTM Technology												
Inverter	16.056	0.029	0.027	2.293	1.530	0.635	-0.655	1.512	2.055	3.581	1.371	2.306
Buffer	18.104	0.018	0.030	3.447	1.311	0.638	-0.167	2.425	3.535	2.753	2.985	1.758
NAND	23.691	0.013	0.008	2.713	1.659	0.769	-0.410	1.232	4.072	3.463	2.283	2.744
NOR	17.761	0.039	0.017	2.398	1.992	0.897	-0.764	6.888	4.936	3.968	8.065	4.895
Inverter Chain	29.972	0.012	0.008	5.205	1.397	0.649	-0.219	2.070	1.448	2.882	3.118	1.907

IV. Conclusion

This paper proposes a simulation and analysis method based on the uniform sampling technique and weighted sample estimator, which requires fewer simulation runs for statistical moment estimation. The number of simulation iterations required by this fast statistical moment estimation (FSME) method is at least two orders of magnitude lower than the number of simulation runs required in the Monte Carlo method. This results into 100× speedup in the SSTA library characterization. Along with this, changes in parameter spread and/or probability density function do not require rerunning of the circuit simulations, which results into faster run time and smaller memory requirement. The state of the art circuit simulation tools can run Monte Carlo with Gaussian, lognormal and uniform distribution only whereas any distribution can be used in the proposed method.

Acknowledgment

The authors would like to thank Prof. Alessandro Di Bucchianico, Eindhoven University of Technology for useful discussions and contributions.

References

[1] C. Forzan and D. Pandini, "Statistical static timing analysis: A survey," *Integration, the VLSI Journal*, vol. 42, no. 3, pp. 409–435, June 2009.

[2] S. Nassif, "Within-chip variability analysis," *IEDM Tech. Dig*, p. 283, 1998.

[3] S. R. Nassif, "Modeling and analysis of manufacturing variations," in *CICC*, 2001, pp. 223–228.

[4] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, "Statistical timing analysis: From basic principles to state of the art," *IEEE TCAD*, vol. 27, no. 4, p. 589, April 2008.

[5] C. Amin, N. Menezes, K. Killpack, F. Dartu, U. Choudhury, N. Hakim, and Y. Ismail, "Statistical static timing analysis: how simple can we get?" in *DAC*. ACM, 2005, p. 657.

[6] C. Visweswariah, K. Ravindran, K. Kalafala, S. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," in *DAC*. ACM, 2004, p. 336.

[7] C. Visweswariah, K. Ravindran, and K. Kalafala, "First-order parameterized block-based statistical timing analysis," *TAU*, 2004.

[8] H. Chang and S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal," in *ICCAD*. IEEE Computer Society Washington, DC, USA, 2003.

[9] R. Rubinstein and D. Kroese, *Simulation and the Monte Carlo method*. Wiley-Interscience, 2007.

[10] P. Glasserman, *Monte Carlo methods in financial engineering*. Springer Verlag, 2003.

[11] Cadence, *Virtuoso Spectre Circuit Simulator Reference*, product version 5.1.41 ed., Cadence Design Systems, Inc., http://www.cadence.com/products/cic/spectre_circuit, November 2004.

[12] L. Zhang, J. Shao, and C. Chen, "Non-gaussian statistical parameter modeling for ssta with confidence interval analysis," in *ISPD*. ACM, 2006, p. 38.

[13] V. Veetil, D. Sylvester, and D. Blaauw, "Criticality aware latin hypercube sampling for efficient statistical timing analysis," *ACM/IEEE TAU*, February 2007.

[14] A. Singhee and R. Rutenbar, "From finance to flip flops: A study of fast quasi-Monte Carlo methods from computational finance applied to statistical circuit analysis," in *ISQED*, 2007.

[15] V. Veetil, D. Sylvester, and D. Blaauw, "Efficient Monte Carlo based incremental statistical timing analysis," in *DAC*, 2008, pp. 676–681.

[16] "Predictive Technology Model (PTM)," Arizona State University, http://ptm.asu.edu, September 2008.

[17] "Open Cell Library, Nangate Inc." http://www.nangate.com, July 2009.