Statistical Moment Estimation of Delay and Power 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

* corresponding author: First Author

Address:

Delft University of Technology

Circuit and System Group

Mekelweg 4

2628 CD, Delft, The Netherlands

Office : +31 15 27 85968

Fax : +31 15 2786190

Email : A.Nigam@tudelft.nl

Statistical Moment Estimation of Delay and Power 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 circuit properties like delay and power. 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 a uniform sampling technique and a weighted sample estimator. The proposed method needs significantly fewer 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 accurate moments estimation of delay and power for standard cells in 45nm and 32nm technologies.

Keywards - Delay, Power, STA, SSTA, Monte Carlo, Statistical Analysis

Statistical Moment Estimation of Delay and Power in Circuit Simulation

I. Introduction

Gate delay and power dissipation are critical issues in present day low power VLSI circuit design. The delay and power of a logic gate strongly depend on variations in process, voltage, and temperature (PVT). As we are moving towards nanometer technology, PVT variations are 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) or power analysis with a corner based approach in advanced technology processes is a serious concern [4]. Due to these PVT variations, delay and power are statistical parameters instead of deterministic ones. The process of estimating the delay of a data path with PVT variation is known as Statistical STA (SSTA) [5–8]. Equivalently, there is statistical power estimation.

In SSTA, the standard cell delay and dynamic power are stochastic parameters, and these 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, standard MC 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 power may result into non-Gaussian distribution of the delay and power. 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 MC 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 parameter sampling in the circuit simulations is still dependent on their distribution and hence not applicable to various types of probability density functions.

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 at least $100\times$ speedup in the library characterization. The application of the FSME method is not limited to digital circuit design and SSTA or power analysis; it is equally applicable to MC simulation in analog circuit design.

The organization of the paper is as follows: The fast statistical moment estimation method is discussed in Section II, followed by simulation results and comparison of FSME and MC in Section III. The conclusion can be found in Section IV.

II. Fast Statistical Moment Estimation Method

The standard MC method is based on 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 start of the simulation. Circuit simulation is repeated for each set of sampled parameters. This results in long run times and high memory requirement to store all the data. The desired simulation output is measured in each simulation, leading to a 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, while using the FSME method, the probability distribution of the process parameters and the circuit simulation are decoupled. In the proposed method, instead of randomly sampling, the space is sampled with a uniform distribution. Moreover, to accurately estimate the statistical moments, a weighted

sample estimator is utilized. The processes involved in circuit simulation and data processing are discussed below.

A. Circuit Simulation

Unlike the 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 length L, channel width W, threshold voltage V_{th}, etc.), where X is a set of vectors X_i, with each vector X_i corresponding to sample 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, power, 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 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 their $pdfs$. 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. Following 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

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

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:

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

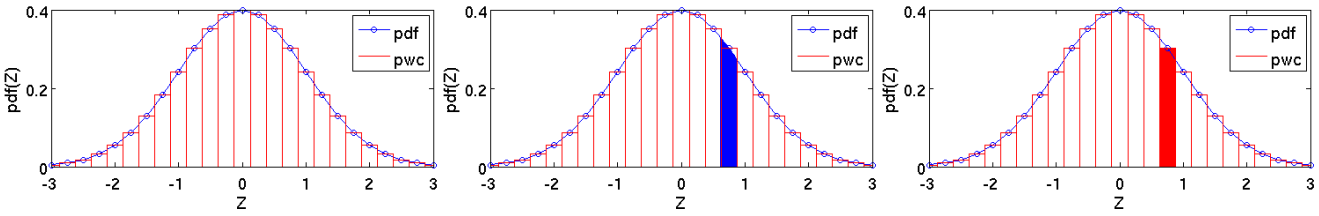
where

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

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*

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

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



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

Fig. 1: Piecewise constant approximation of probability density function

integration.

To illustrate this with an example, consider the 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 have opposite signs and this neutralization effect reduces the error due to the approximation. The total error can be reduced by increasing the 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

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

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. 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 the 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

$$P_s(k) = P_d(X_1[j_1]) \cdot P_d(X_2[j_2]) \dots \tag{9}$$

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:

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

$$\Rightarrow P_d(Y[k]) \ = P_d(X_1[j_1]) \cdot P_d(X_2[j_2]) \dots \tag{11}$$

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 $P_d(Y[k])$ is not the probability of $Y = Y[k]$, as more than one experiment could produce the same output value $Y[k]$ in case of a non-monotonous function of Y. Each $P_d(Y[k])$ will have a 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 $\lceil N \cdot P_d(Y[k]) \rceil$ 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 vector $R(Y[k])$ as the set of experiment outputs $Y[k]$ which repeats $\lceil N \cdot P_d(Y[k]) \rceil$ times, i.e.

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

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

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

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

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

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

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

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

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

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 appears in both the numerator as well as denominator, and it cancels out. 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. This is very important, as in reality the $pdfs$ of parameter variations change while a manufacturing process matures.

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 based Nangate technology file [17]. The process variations are considered to be 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 space limitation, only two sets of variations are discussed here. In first set, the variation considered is in the effective channel length (L) of the MOSFET with $3\sigma_L$ equal to 20% of the nominal value of L. In addition to the first set, variation in the effective channel width (W) is also considered in the second set with $3\sigma_W$ 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 and dynamic power (rising edge) 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 four moments of the dynamic power vs simulation runs for the 45nm inverter with first set of variations are shown in Figures 2b. Additionally, these first four moments of the delay and dynamic power vs simulation runs for the 32nm inverter with second set of variations are shown in Figures 3a and Figures 3b respectively. 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 in the statistical moment estimation of the delay for the first set of variation in the 45nm inverter (see Figures 2a) 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 0.15% whereas fifty iterations of FSME give only 0.03% of error. Similarly, the error for standard deviation estimation using MC is 0.40% whereas the FSME has an error of only 0.26%. The MC error in skewness estimation is 3.50% and FSME gives an error of only 0.97%. Lastly, kurtosis estimation has an error of 3.64% in MC where as FSME is at 1.28% error margin.

Similar to the delay, the error in the statistical moment estimation of the dynamic power for the first set of variation in the 45nm inverter (see Figures 2b) with respect to the respective reference value is reported below. It can be seen form these plots that the convergence of the power estimation needs more simulation iterations. Therefore, ten thousand iterations in MC and hundred iterations in FSME are used for the error comparison. The error in the mean estimation using ten thousand iterations of MC with reference value is 0.01% whereas one hundred iterations of FSME also has only 0.01% of error. Similarly, the error for standard deviation estimation using MC is 7.38% whereas the FSME has an error of only 2.10%. The MC error in skewness estimation is 7.67% and FSME gives an error of only 3.47%. Lastly, kurtosis estimation has an error of 10.35% in MC where as FSME is at 7.48% 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 in the statistical moment estimation of the delay (see Figures 3a) with the respective reference value. The error in the mean estimation of the delay using ten thousand iterations of MC with reference value is 0.07% whereas hundred iterations of FSME has only 0.03% of error. Similarly, the error for standard deviation estimation using MC is 0.18% whereas the FSME has an error of 0.69%. The MC error in skewness estimation is 2.26% and FSME gives an error of 3.82%. Lastly, kurtosis estimation has an error of 3.02% in MC where as FSME is at 4.03% error margin.

The error in the statistical moment estimation of the dynamic power for the second set of variation in the 32nm inverter (see Figures 3b) with respect to the respective reference value is reported below. In the error estimation for the second set of variation, forty thousand iterations in MC and four hundred iterations in FSME are used for the error comparison. The error in the mean estimation using forty thousand iterations of MC with reference value is 0.07% whereas four hundred iterations of FSME has only 0.06% of error. Similarly, the error for standard deviation estimation using MC is 14.81% whereas the FSME has an error

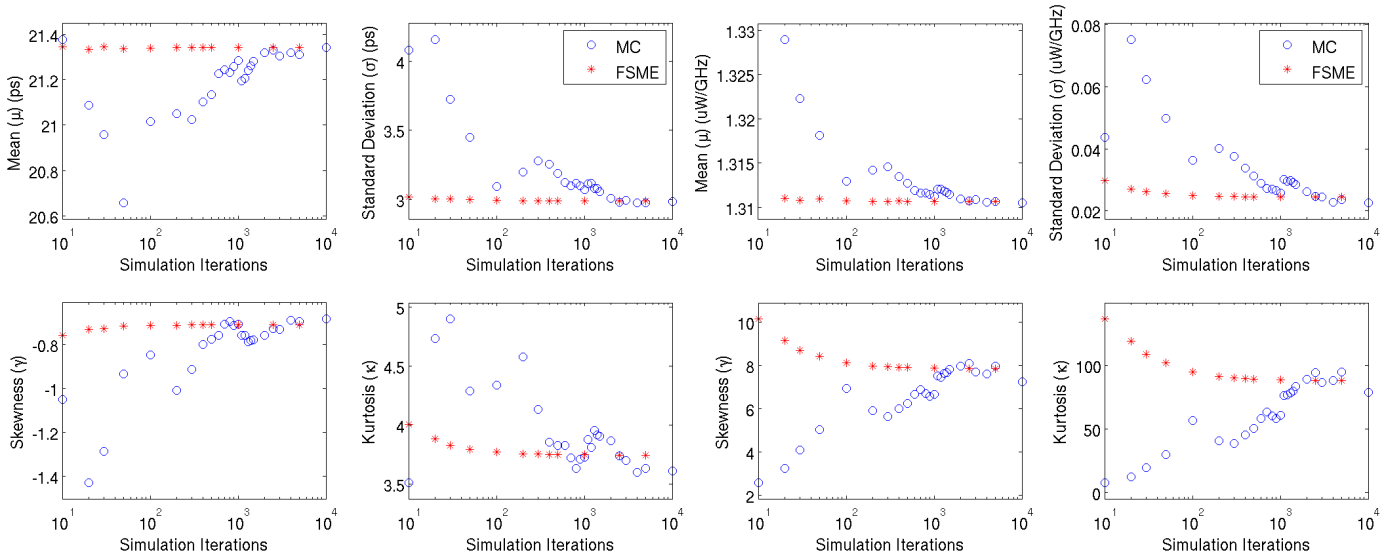
of 15.69%. The MC error in skewness estimation is 7.74% and FSME gives an error of only 8.38%. Lastly, kurtosis estimation has an error of 5.99% in MC where as FSME is at 8.63% error margin.

It is clear from above experimental results that MC with five thousand iterations produce more inaccurate results for moment estimation of the delay in comparison to the respective golden reference than FSME with only fifty simulations for one parameter variations. Similarly, only hundred iterations of FSME produce more accurate statistical moment estimation for the dynamic power measurement as compared with ten thousand MC iterations for one parameter variation. For two parameter variations, MC with ten thousand iterations has almost equal error for delay variation estimation in comparison to the respective golden reference than FSME with only one hundred iterations. Additionally, forty thousand iterations of MC produce almost equal error as compare to the four hundred iterations of the FSME for spread measurement in the dynamic power estimation.

A similar behaviour is observed in all the five test circuits in both the 45nm and the 32nm technologies for statistical moment estimation of delay and dynamic power. The reference value of these four moments along with the error in the moment estimation of the delay for MC with five thousand runs and FSME with fifty runs using 45nm and 32nm technology with the first variation set are reported in Table I. A similar table for the second set of variations is reported in Table II. The error in the moment estimation of the dynamic power using ten thousand iterations of MC and one hundred iterations of FSME for 45nm and 32nm for the first set of variation set are reported in Table III. Similarly, the error in the dynamic power estimation for the second set of variations is reported in Table IV. The plots of the moment estimation of the delay and dynamic power vs simulation runs for Inverter in 32nm PTM using the first set of variations are shown in Figures 4a and 4b, 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 a 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 a 45nm buffer using these probability density functions are shown in Figure 5. 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 that the higher order moments significantly vary with the distribution of the parameters.

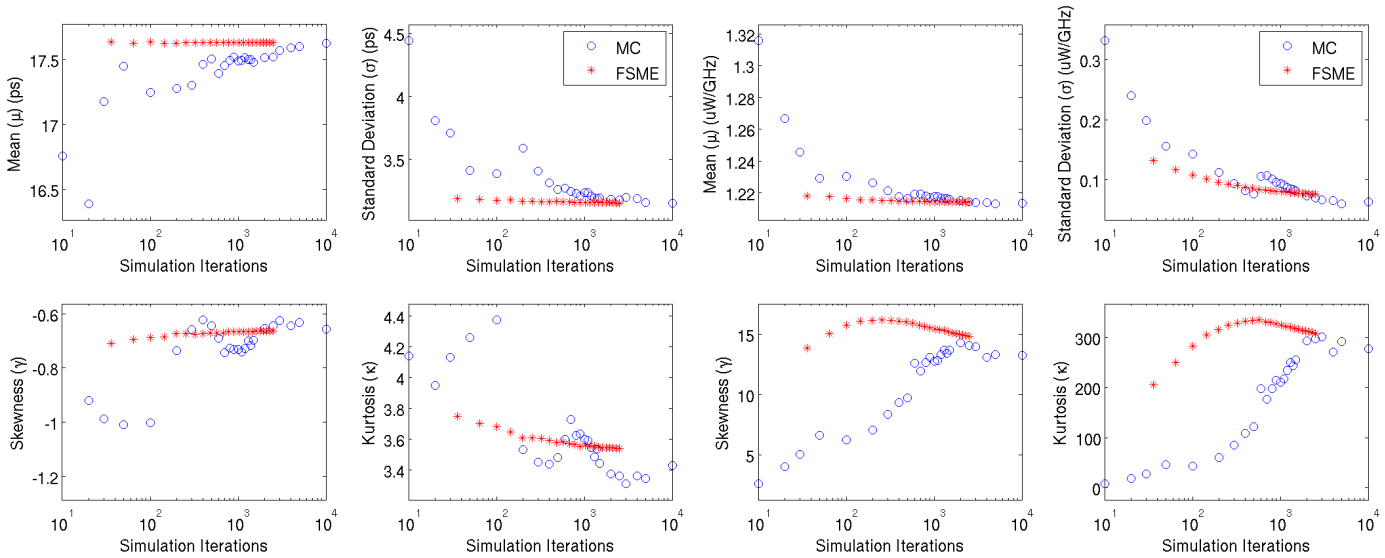
The skewness and kurtosis of the delay in above result indicates that the distribution of the delay does



(a) 45nm Inverter - Delay - One parameter variation

(b) 45nm Inverter - Power - One parameter variation

Fig. 2: The first four moment estimation of delay and power vs simulation iterations for MC and FSME with one parameter (L) variation in 45nm Inverter



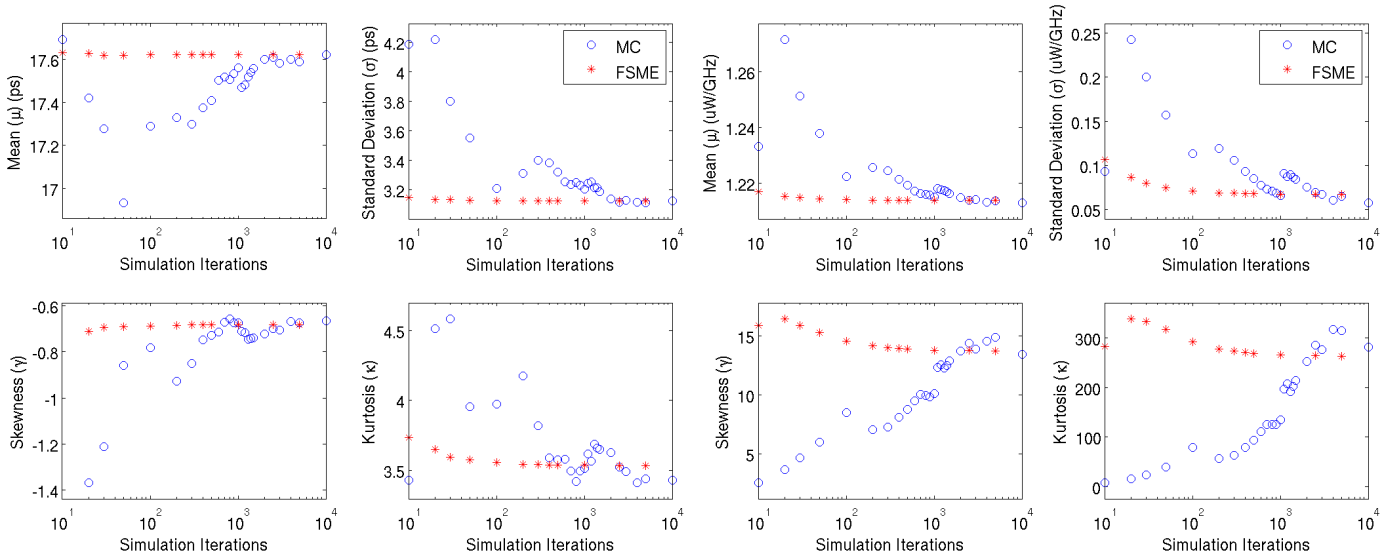
(a) 32nm Inverter - Delay - Two parameter variations

(b) 32nm Inverter - Power - Two parameter variations

Fig. 3: The first four moment estimation of delay and power vs simulation iterations for MC and FSME with two parameters (L and W) variations in 32nm Inverter

not follow a Gaussian distribution. However, the distribution of the delay variation is close to a Gaussian distribution. In contrast to the delay, the skewness and kurtosis of the dynamic power is very high. It shows that the distribution of the dynamic power dissipation is significantly deviated from a Gaussian distribution.

The above results show that the simulation iterations required in FSME to estimate the moments



(a) 32nm Inverter - Delay - One parameter variation

(b) 32nm Inverter - Power - One parameter variation

Fig. 4: The first four moment estimation of delay and power vs simulation iterations for MC and FSME with one parameter (L) variation in 32nm Inverter

TABLE I: Error % comparison in the first four moments estimation of delay (ps) 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**	**MC**	**FSME**	**Ref**	**MC**	**FSME**	**Ref**	**MC**	**FSME**	**Ref**	**MC**	**FSME**
45nm PTM Technology												
Inverter	21.34	0.15	0.03	2.99	0.40	0.26	-0.71	3.50	0.97	3.75	3.64	1.28
Buffer	21.58	0.18	0.01	3.34	0.43	0.18	-0.31	2.66	1.05	2.96	1.80	0.74
NAND	22.22	0.15	0.00	3.13	0.37	0.25	-0.77	3.68	1.62	3.85	4.13	1.68
NOR	32.83	0.13	0.01	3.84	0.54	0.28	-0.39	5.03	1.72	3.29	2.66	1.15
Inverter	41.16	0.18	0.01	6.44	0.50	0.24	-0.29	4.07	1.45	3.02	1.94	0.82
32nm PTM Technology												
Inverter	17.62	0.19	0.01	3.12	0.31	0.26	-0.68	2.63	1.21	3.53	2.97	1.14
Buffer	18.10	0.22	0.02	3.42	0.44	0.24	-0.19	3.84	1.32	2.71	1.43	0.40
NAND	18.15	0.19	0.03	3.24	0.37	0.32	-0.74	3.54	1.08	3.78	3.99	1.38
NOR	30.16	0.18	0.00	4.65	0.58	0.33	-0.27	5.72	0.85	3.10	2.07	0.99
Inverter	31.85	0.23	0.01	6.25	0.49	0.20	-0.23	3.53	0.88	2.89	1.71	0.67

differ from the simulation iterations required in MC by two orders of magnitude. Furthermore, different parameter spread can be analyzed in FSME by only changing the parameters 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 II: Error % comparison in the first four moments estimation of delay (ps) 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**	**MC**	**FSME**	**Ref**	**MC**	**FSME**	**Ref**	**MC**	**FSME**	**Ref**	**MC**	**FSME**
45nm PTM Technology												
Inverter	21.36	0.06	0.01	3.03	0.22	0.78	-0.70	2.86	5.09	3.76	3.76	4.92
Buffer	21.60	0.06	0.02	3.38	0.10	0.70	-0.30	1.30	2.95	2.98	1.94	1.92
NAND	22.23	0.06	0.03	3.17	0.27	0.84	-0.76	3.23	5.74	3.87	4.26	6.44
NOR	32.86	0.05	0.01	3.90	0.20	0.79	-0.37	3.10	5.83	3.31	3.13	3.99
Inverter	41.18	0.06	0.02	6.47	0.16	0.71	-0.29	2.74	3.59	3.04	2.20	2.84
32nm PTM Technology												
Inverter	17.63	0.07	0.03	3.16	0.18	0.69	-0.66	2.26	3.82	3.54	3.02	4.03
Buffer	18.12	0.07	0.03	3.45	0.08	0.51	-0.17	4.91	3.30	2.74	1.51	2.07
NAND	18.16	0.08	0.03	3.27	0.30	0.95	-0.73	3.05	6.27	3.81	4.06	5.95
NOR	30.20	0.06	0.01	4.73	0.22	0.73	-0.23	2.79	9.12	3.12	2.55	3.34
Inverter	31.87	0.09	0.02	6.28	0.11	0.63	-0.23	2.34	2.25	2.91	1.88	2.31

TABLE III: Error % comparison in the first four moments estimation of dynamic power (μW/GHz) for one parameter (L) 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**	**MC**	**FSME**	**Ref**	**MC**	**FSME**	**Ref**	**MC**	**FSME**	**Ref**	**MC**	**FSME**
45nm PTM Technology												
Inverter	1.31	0.01	0.01	0.02	7.38	2.10	7.85	7.67	3.47	88.06	10.35	7.48
Buffer	1.26	0.02	0.01	0.04	7.01	2.15	7.03	9.98	4.34	78.35	13.05	8.89
NAND	1.31	0.01	0.00	0.02	7.36	2.50	7.47	9.03	3.21	84.88	11.66	6.93
NOR	1.48	0.01	0.00	0.01	5.02	1.92	3.23	26.29	19.47	39.17	28.20	31.15
Inverter	2.46	0.02	0.01	0.05	10.76	3.48	10.37	5.80	4.80	153.57	3.33	9.67
32nm PTM Technology												
Inverter	1.21	0.06	0.02	0.07	14.10	5.44	13.74	1.98	5.97	263.05	7.13	11.30
Buffer	1.20	0.26	0.08	0.22	18.72	8.53	19.43	8.95	4.95	486.06	30.88	8.80
NAND	1.22	0.05	0.02	0.06	14.65	5.44	14.27	0.60	5.22	277.66	9.57	9.85
NOR	1.32	0.08	0.02	0.07	19.46	8.20	20	12.11	3.57	494.79	35.53	6.63
Inverter	2.00	0.35	0.12	0.47	20.30	9.80	21.72	14.16	4.23	588.56	41.50	7.46

IV. Conclusion

This paper proposes a simulation and analysis method based on a uniform sampling technique and a 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$\times$ 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,

TABLE IV: Error % comparison in the first four moments estimation of dynamic power (μW/GHz) for two parameters (L and W) variation using Monte Carlo (40000 runs) and proposed method (400 runs) in the 45nm and 32nm PTM technology

	Mean (μ)			Standard Deviation (σ)			Skewness (γ)			Kurtosis (κ)		
Circuits	**Ref**	**MC**	**FSME**	**Ref**	**MC**	**FSME**	**Ref**	**MC**	**FSME**	**Ref**	**MC**	**FSME**
45nm PTM Technology												
Inverter	1.31	0.01	0.01	0.03	4.92	4.62	5.00	16.31	13.83	53.06	23.21	23.21
Buffer	1.26	0.01	0.02	0.05	6.30	5.76	6.73	13.84	12.91	79.04	20.41	22.44
NAND	1.31	0.01	0.01	0.03	4.39	4.01	3.89	19.15	17.36	39.42	25.86	27.37
NOR	1.48	0.01	0.01	0.03	1.50	1.47	0.27	45.91	58.44	4.32	18.02	29.31
Inverter	2.46	0.03	0.02	0.10	2.98	2.94	1.56	31.67	33.88	15.90	32.09	41.35
32nm PTM Technology												
Inverter	1.21	0.07	0.06	0.08	14.81	15.69	14.83	7.74	8.38	308.65	5.99	8.63
Buffer	1.21	0.29	0.29	0.26	20.74	23.33	21.08	2.02	1.53	555.79	12.40	8.5
NAND	1.22	0.06	0.05	0.06	14.92	15.35	14.92	6.91	8.02	310.40	4.18	7.79
NOR	1.32	0.09	0.08	0.08	19.74	21.32	19.89	1.08	2.42	506.17	11.47	4.98
Inverter	2.00	0.40	0.41	0.57	22.45	25.85	22.93	5.05	2.62	642.99	18.23	13.48

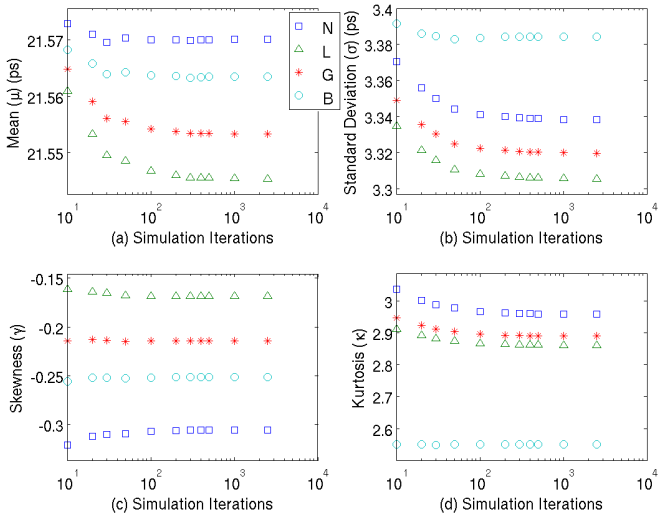


Fig. 5: Moment estimation vs simulation run in 45nm Buffer using Gaussian (N), Lognormal (L), Gamma (G), and Beta (B) distributions.

which results into faster run time and smaller memory requirement. 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," *International Electron Devices Meeting Technical Digest*, p. 283, 1998.

[3] S. R. Nassif, "Modeling and analysis of manufacturing variations," in *IEEE Conference on Custom Integrated Circuits*, 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 Transactions on Computer Aided Design of Integrated Circuits and Systems*, 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 *Design Automation Conference*. 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 *Design Automation Conference*. ACM, 2004, p. 336.

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

[8] H. Chang and S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal," in *International Conference on Computer-Aided Design*. 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 *International Symposium on*

Quality Electronic Design, 2007.

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

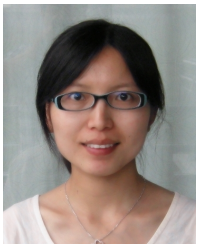
[16] "Predictive Technology Model (PTM)," Arizona State University, http://ptm.asu.edu, September 2008.

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

BIOGRAPHIES

Ashish Nigam received the B.Tech. degree in the department of Electrical Engineering from Indian Institute of Technology, Kanpur, India in 2005 and the M.Sc. Degree in the department of Microelectronics and Computer Engineering from Delft University of Technology, Delft, The Netherlands in 2010. From 2005 to 2008, he was design engineer at Intel in Bangalore, India, working on the integrated graphics and memory controllers. In 2010, he joined Delft University of Technology as a researcher in the Circuit and Systems Group.

His research interest include digital circuit design, process variations, gate modeling, statistical static timing analysis and its methodology development.



Qin Tang received the B.Sc. degree in the department of Electronic Engineering in 2006, and the M.Sc. degree in the department of Microelectronics and Solid State Electronics, school of Electronic Science and Engineering in 2008, from Southeast University, Nanjing, China. She is currently working toward the Ph.D. degree in the Circuits and Systems group, the department of Microelectronics and Computer Engineering, Delft University of Technology.

Her research interests include the analysis of process variations, gate modeling for timing analysis and statistical static timing analysis.

Amir Zjajo (M'02) received the M.Sc. and DIC degrees from the Imperial College London, London, U.K., in 2000 and the Ph.D. degree from Eindhoven University of Technology, Eindhoven, The Netherlands in 2010, all in electrical engineering. In 2000, he joined Philips Research Laboratories as a member of the research staff in the Mixed-Signal Circuits and Systems Group. From 2006 until 2009, he was with Corporate Research of NXP Semiconductors as a senior research scientist. In 2009, he joined Delft University of Technology as a Faculty member in the Circuit and Systems Group.

Dr. Zjajo has published more than 30 papers in referenced journals and conference proceedings, and holds 3 patents with 7 more pending. He is the author of the book Low-Voltage High-Resolution A/D Converters: Design and Calibration (Springer, 2010). He serves as a member of Technical Program Committee of IEEE Design, Automation and Test in Europe Conference, and IEEE International Mixed-Signal Circuits, Sensors and Systems Workshop. His research interests include mixed-signal circuit design, signal integrity and timing and yield optimization of VLSI.

Michel Berkelaar (M'96) received the M.Sc. and Ph.D. degrees in Electrical Engineering from Eindhoven University of Technology in 1987 and 1992 respectively. From 1992 to 2000 he was assistant and associate professor at Eindhoven University of Technology. From 2000 to 2009 he was director of research at Magma Design Automation in Eindhoven, the Netherlands. Currently, he is project leader for the European ENIAC MODERN project at Delft University of Technology.

Dr. Berkelaar has published more than 50 papers at conferences and in journals, mainly in the areas of logic synthesis and statistical timing. He is an active reviewer and occasional topic chair for Design Automation and Test Europe (DATE), the International Conference on Computer-Aided Design (ICCAD) and the International Workshop on Logic and Synthesis (IWLS). For the latter he has also served as program chair.

Nick van der Meijs (M'87) received M.Sc and Ph.D degrees from Delft University of Technology in 1985 and 1992 respectively. Currently, he is associate professor at Delft University of Technology in Delft, the Netherlands in the Circuits and Systems group of the Department of Micro Electronics and Computer Engineering.

He has (co-)authored some 100 papers on various topics including design frameworks, interconnect optimization and parasitics modeling, and was one of the lead developers of the SPACE 2D and 3D parasitic layout to circuit extractor. He and his research group currently work both on modeling of parasitic effects in advanced integrated circuits and on circuit level design methods and tools for dealing with variability. He is a regular reviewer for various EDA and design methodology conferences and journals, and has served as topic chair on multiple at conferences. As a Director of Studies he is responsible for the content, organization and quality of the BSc and MSc curricula in Electrical Engineering and Computer Engineering at TU Delft.