Fast Monte Carlo Method via Reduced Sample Number and Node Filtering

Inhak Han, Lee-eun Yu, and Youngsoo Shin
Department of Electrical Engineering
KAIST, Daejeon 305-701, Korea

***Abstract*— Monte Carlo (MC) method is convenient and robust to estimate timing yield of circuits under the influence of process variations. The important question in MC method is the number of samples while we assure a desired accuracy of yield estimate, which is often addressed using a rule of thumb. Minimum number of samples can be estimated via approximation by a normal distribution, but the provided number may be too small to be used in practice considering that target yield, which is used to derive the number, is unknown. Chebyshev's inequality has been used to derive a sample number, but the number is too large this time. We develop a new expression, which provides the sample number that is much closer to the minimum (3× to 8×) compared to the number provided by Chebyshev's inequality (5× to 15×). We also propose a simple node filtering algorithm, where we identify the nodes that are likely to affect timing yield; the simulation with each MC sample can handle only a fraction of circuit elements as a result. Reducing the number of MC samples and simulating only selected nodes together yield 27× to 125× speedup over standard MC method.**

I. Introduction

Within-die (WID) variations account for an increasing proportion of the total process variations: e.g. 35% in 130 nm technology, but 60% in 70 nm technology [1]. To incorporate WID variations in timing analysis, each gate delay has to be modeled as a random variable. Timing analysis can then be performed using either Monte Carlo (MC) method or statistical static timing analysis (SSTA). MC method relies on repeated random sampling of the value of each random gate delay, followed by traditional static timing analysis (STA). SSTA [2] is an extension of STA; whereas STA propagates simple arrival times (ATs) and required arrival times (RATs), SSTA propagates random variables that describe ATs and RATs.

MC method or SSTA can be used to compute a timing yield, which is the probability that a circuit satisfies given timing constraints. SSTA is often claimed to be much faster than MC method; MC method in fact has been used as a reference to assess the accuracy and runtime of SSTA. The standard MC method to timing yield analysis is indeed very slow. But this can be alleviated in two directions, which is the focus of this paper: by reducing the number of random samples and by reducing the number of nodes that should be simulated.

Variance reduction techniques have been used to reduce the variance of yield estimate without increasing sample numbers. These include quasi Monte Carlo method, Latin hypercube sampling, orthogonal sampling, stratified sampling, control variates, and so on. But, these techniques have their own limitations: random variables and yield should be correlated, some techniques require a few random variables to be selected, some are very hard to implement, some require a lot of memory space, etc.

If we stick to use MC method, which is convenient and robust, the foremost question is to know how many samples are required (or how many simulations have to be performed) to achieve the desired accuracy of yield estimate. Unfortunately, this has been done in a rule of thumb basis; 10k or 1M are widely used numbers. Minimum number of samples can be estimated via approximation by a normal distribution, but the provided number may be too small to be used in practice considering that target yield, which is used to derive the number, is unknown. Chebyshev's inequality [3] has been used to derive a sample number, but the number is too large this time. We develop a new expression (Section II), which provides the sample number that is much closer to the minimum (3× to 8×) compared to the number provided by Chebyshev's inequality (5× to 15×).

Our second tactic to reduce the runtime of MC method is to simulate a fraction of the circuit elements (Section III). This is suggested by the observation that the paths likely to become critical usually include only small proportion of the total number of gates, say 1% [4]. We try a simple heuristic algorithm (Section III) for this purpose. It first runs a STA to pick the primary output that trails the critical path; it is then used to select a few more primary outputs that are likely to be critical under process variations. Primary inputs that can be reached from these primary outputs are found out. Only the gates that are within a region spanned by these primary inputs and outputs are marked and simulated. Experiments show that 9% of gates are marked on average under max-delay constraint and 4% under min-delay constraint.

II. Sample Number of MC Method

Let y be a timing yield, which is unknown. We want to estimate y with confidence of c, such that an estimated value $\hat{y}$ lies in the interval $[y(1-\varepsilon),\ y(1+\varepsilon)]$, i.e.

$$\Pr\left(y(1-\varepsilon) \le \hat{y} \le y(1+\varepsilon)\right) \ge c, \qquad (1)$$

or

$$\Pr\left(|\hat{y}-y| \le y\varepsilon\right) \ge c. \qquad (2)$$

The question we want to answer is the number of samples N of MC method such that (2) is satisfied.

After simulating a circuit with each MC sample, a circuit passes or fails (to meet timing constraints) with a probability of y. Let X_i be the outcome of i-th simulation, i.e. $X_i = 1$

corresponds to pass and $X_i = 0$ otherwise. Let $\sum_{i=1}^{N} X_i$ be denoted by a random variable X. Since X follows a binomial distribution $B(N, y)$, its mean and variance are Ny and $Ny(1-y)$, respectively. The estimated timing yield $\hat{y}$ is then the average of such outcomes after N simulations, i.e. $\hat{y} = X/N$.

A. Sample Number from Normal Approximation

For sufficiently large N, X can be approximated by a normal distribution due to central limit theorem, i.e. $X \sim N(Ny, Ny(1-y))$; $\hat{y}$ also follows a normal distribution, $\hat{y} \sim N(y, y(1-y)/N)$. Therefore,

$$\frac{\hat{y}-y}{\sqrt{(y(1-y)/N)}} \sim N(1,0). \tag{3}$$

Now we want to find the sample number N that satisfies (2), or equivalently

$$\Pr\left(\left|\frac{\hat{y}-y}{\sqrt{(y(1-y)/N)}}\right| \leq \frac{y\varepsilon}{\sqrt{(y(1-y)/N)}}\right) \geq c. \tag{4}$$

Using standard normal cumulative distribution function $\Phi(\cdot)$ yields

$$\Phi\left(\frac{y\varepsilon}{\sqrt{(y(1-y)/N)}}\right) - \Phi\left(-\frac{y\varepsilon}{\sqrt{(y(1-y)/N)}}\right) \geq c. \tag{5}$$

Since $\Phi(-x)$ is equal to $1-\Phi(x)$, (5) is equivalent to

$$2\Phi\left(\sqrt{\frac{Ny\varepsilon^2}{1-y}}\right) - 1 \geq c. \tag{6}$$

Solving for N yields

$$N = N_{nor} \geq \frac{1-y}{y\varepsilon^2}\left(\Phi^{-1}\left(\frac{c+1}{2}\right)\right)^2. \tag{7}$$

B. Sample Number Using Chebyshev's Inequality

Chebyshev's inequality has been used to derive a sample number [3]. Chebyshev's inequality states that

$$\Pr(|X-\mu| \leq \alpha) \geq 1 - \frac{\sigma^2}{\alpha^2}, \tag{8}$$

where μ and σ^2 are mean and variance of X, respectively, and α is any positive real number. Substituting $\mu = Ny$ and $\sigma = Ny(1-y)$, and choosing $Ny\varepsilon$ for α yields

$$\Pr(|y - X/N| \leq y\varepsilon) \geq 1 - \frac{1-y}{Ny\varepsilon^2}. \tag{9}$$

The right-hand side has to be greater than or equal to c to satisfy (2); solving that constraint for N yields

$$N = N_{che} \geq \frac{1-y}{(1-c)y\varepsilon^2}. \tag{10}$$

C. The New Tighter Sample Number

Since the sample number from (10) is too large causing an unnecessary amount of large simulation time as we show in Section II-D, we develop a new closed-form expression for N.

Inequality (2) is equivalent to

$$\Pr(|y-\hat{y}| \geq y\varepsilon) \leq 1-c. \tag{11}$$

Substituting X/N for $\hat{y}$ lets the left-hand side be transformed to

$$\Pr(|X-Ny| \geq Ny\varepsilon). \tag{12}$$

We first consider when $X - Ny > 0$. For some $s > 0$,

$$\Pr(X - Ny \geq Ny\varepsilon) = \Pr\left(e^{s(X-Ny)} \geq e^{sNy\varepsilon}\right), \tag{13}$$

which is smaller than or equal to

$$\frac{\mathrm{E}\left(e^{s(X-Ny)}\right)}{e^{sNy\varepsilon}} = \frac{\prod_{i=1}^{N}\mathrm{E}\left(e^{s(X_i-y)}\right)}{e^{sNy\varepsilon}} = e^{-sNy\varepsilon}\left[\mathrm{E}\left(e^{s(X_i-y)}\right)\right]^N \tag{14}$$

by Markov's inequality. The upper bound of $\mathrm{E}\left(e^{s(X_i-y)}\right)$ can be determined by the manipulation via Taylor series expansion, and is given by

$$\mathrm{E}\left(e^{s(X_i-y)}\right) \leq e^{\frac{s^2}{2}y(1-y)}, \tag{15}$$

which holds when $y \geq 0.5$. The details of the derivation is omitted. Combining (15) with (13) yields

$$\Pr(X - Ny \geq Ny\varepsilon) \leq e^{-sNy\varepsilon + \frac{s^2}{2}Ny(1-y)} = e^{-\frac{Ny\varepsilon^2}{2(1-y)}} \tag{16}$$

by choosing $s = \frac{\varepsilon}{1-y}$, which is positive.

When $X - Ny < 0$, we can follow the similar procedure to show

$$\Pr(X - Ny \leq -Ny\varepsilon) \leq e^{-2Ny^2\varepsilon^2}. \tag{17}$$

The details are again omitted. Merging (16) and (17) gives us

$$\Pr(|X-Ny| \geq Ny\varepsilon) \leq e^{-\frac{Ny\varepsilon^2}{2(1-y)}} + e^{-2Ny^2\varepsilon^2} \leq 1-c. \tag{18}$$

We force, in (18), that

$$e^{-\frac{Ny\varepsilon^2}{2(1-y)}} \leq e^{-4Ny^2\varepsilon^2}, \tag{19}$$

which allows

$$e^{-\frac{Ny\varepsilon^2}{2(1-y)}} + e^{-2Ny^2\varepsilon^2} \leq \left(e^{-2Ny^2\varepsilon^2}\right)^2 + e^{-2Ny^2\varepsilon^2} \leq 1-c. \tag{20}$$

Solving (20) for N gives us

$$N = N_{new} \geq \frac{\ln(-\frac{1}{2} + \sqrt{\frac{5}{4} - c})}{2y^2\varepsilon^2}, \tag{21}$$

which holds when $y \gtrapprox 0.854$, derived from solving (19) for y. The constraint on the value of y is a limitation of the new closed-form expression (20), which however is not very bad since timing yield is typically maintained not too far from 1.0

TABLE I

COMPARING SAMPLE NUMBER OF MC METHOD: NUMBER FROM NORMAL APPROXIMATION N_{nor}, NUMBER USING CHEBYSHEV'S INEQUALITY N_{che}, AND THE NUMBER FROM OUR NEW INEQUALITY N_{new}

		y=0.86			y=0.90			y=0.95		
c	ε	N_{nor}	N_{che}	N_{new}	N_{nor}	N_{che}	N_{new}	N_{nor}	N_{che}	N_{new}
0.95	0.005	25000	130233	82271	17000	88889	75120	8088	42106	67421
	0.010	6254	32559	20568	4270	22230	18780	2022	10527	16856
	0.015	2780	14471	9142	1897	9877	8347	899	4679	7492
0.99	0.005	43200	651163	124798	29500	444445	113952	13970	210527	102272
	0.010	10803	162791	31200	7374	111112	28488	3493	52632	25568
	0.015	4802	72352	13867	3277	49383	12662	1553	23392	11364

D. Numerical Analysis

Table I shows sample numbers for two different confidence levels c, 0.95 and 0.99. For each confidence level, we try three different values of ε (0.005, 0.01, and 0.015) and three different values of target yield (0.86, 0.90, and 0.95). The sample number from new inequality N_{new} and that from Chebyshev's inequality N_{che} are shown as compared to N_{nor}, which is considered to represent the minimum sample number.

It can be shown that both N_{new}/N_{nor} and N_{che}/N_{nor} are almost independent of ε. From (10) and (21), we can see that N is proportional to $1/\varepsilon^2$ in both inequalities; this suggests that N_{nor} will also be proportional to $1/\varepsilon^2$, which is in agreement with (7). N_{che}/N_{nor} is weakly dependent on y, which suggests that N_{nor} will be proportional to $(1-y)/y$, which N_{che} is proportional to. Since N_{new} is proportional to $1/y^2$ from (21), N_{new}/N_{nor} should then be proportional to $1/y(1-y)$, which matches very well with the numbers reported in columns 5, 8, and 11 of Table I. In this regard, we can conclude that N_{che} represents the property of minimum sample number N_{nor} for y and ε. On the other hand, as c increases, N_{new}/N_{nor} tends to decrease, which is positive property especially because sample number itself increases with increasing c. If we can combine positive aspects of N_{che} (in its representing the property of N_{nor} for y and ε) and N_{new} (in its insensitivity to c), we could find a better closed-form expression for N, which is left for future work.

In (9), the lower bound of probability (the probability that the difference of yield estimate and target yield lies in the interval $[-y\varepsilon, y\varepsilon]$) is in a form of $1-\alpha/N$ for N_{che}; the corresponding lower bound for N_{new} is in a form of $1-e^{-\beta N}$. As c increases, we need more samples N. As N increases, $1-e^{-\beta N}$ becomes closer to 1 faster than $1-\alpha/N$ does; equivalently, the increase of N_{che} will be faster than that of N_{new} if we keep α/N_{che} and $e^{-\beta N_{new}}$ in the same amount, which explains faster increase of N_{che} compared to N_{new} as c changes its value from 0.95 to 0.99.

Note that N_{new} is not smaller than N_{che} when $y=0.95$ and $c=0.95$; nevertheless, this is not too bad since it is the case when sample number can be kept small compared to other combinations of y and c.

III. NODE FILTERING

We propose a simple node filtering algorithm, so that each simulation can handle only a fraction of circuit elements.

Algorithm *Node_Filtering*

L1 **for** each of WC, NC, and BC **do**
L2 Perform STA
Setup time constraint:
L3 $M_j \leftarrow$ maximum AT at each output j
L4 $M^* \leftarrow \max_j M_j$
L5 $O \leftarrow \{j \mid M^* - M_j \leq M^*\rho_s\}$; $O_{\mathcal{S}} \leftarrow O_{\mathcal{S}} \cup O$
L6 $I_{\mathcal{S}} \leftarrow I_{\mathcal{S}} \cup \{i \mid i$ is reachable from $j \in O, s_i \leq M^*\rho_s\}$
Hold time constraint:
L7 $m_j \leftarrow$ minimum AT at each output j
L8 $m^* \leftarrow \min_j m_j$
L9 $O' \leftarrow \{j \mid m_j - m^* \leq m^*\rho_h\}$; $O_{\mathcal{H}} \leftarrow O_{\mathcal{H}} \cup O'$
L10 $I_{\mathcal{H}} \leftarrow I_{\mathcal{H}} \cup \{i \mid i$ is reachable from $j \in O', s_i \leq m^*\rho_h\}$
end
L11 $\mathcal{G}_{\mathcal{S}} \leftarrow$ Nodes within a region spanned by $I_{\mathcal{S}}$ and $O_{\mathcal{S}}$
L12 $\mathcal{G}_{\mathcal{H}} \leftarrow$ Nodes within a region spanned by $I_{\mathcal{H}}$ and $O_{\mathcal{H}}$

Fig. 1. Algorithm of *Node_Filtering*.

A. Algorithm

Fig. 1 shows a heuristic algorithm. We first perform STA (L2), i.e. perform timing analysis without assuming any WID variations. For each output j, which is either flip-flop input or circuit output, its maximum arrival time M_j is obtained (L3). The maximum value of M_j, which corresponds to the output associated with timing-critical path, is called M^* (L4). The output j that satisfies $M^* - M_j \leq M^*\rho_s$ for some constant ρ_s is considered to be a candidate for simulation with each MC sample, thus is marked (L5). If we repeat simulations with different MC samples, M_j and M^* will eventually be associated with probability distributions. Unless the distribution of M_j is sufficiently close to that of M^*, the output j is unlikely to affect timing yield, which forms the motivation of L5. We propagate the RATs of candidate outputs toward the inputs, and compute slacks of the inputs that can be reached from the candidate outputs. Out of those inputs, we mark input i if its slack s_i is sufficiently small, which is assessed by $s_i \leq M^*\rho_s$ (L6); this is based on the fact that an input is likely to affect timing yield only if it has a very small slack.

A similar procedure is performed to mark outputs and inputs that are likely to affect timing yield under hold time constraint (L7 to L10). The process is repeated under different process corners (L1), e.g. worst corner (WC), nominal corner (NC),

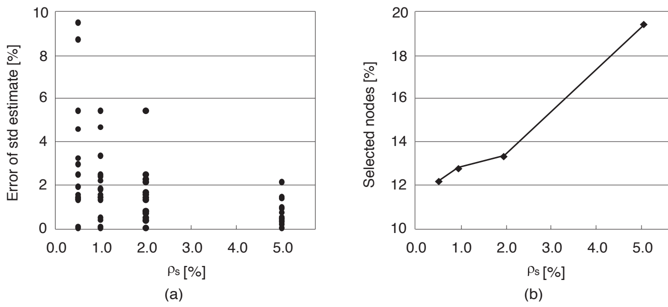

Fig. 2. MC method from N_{che} samples with *Node_Filtering*: (a) error of standard deviation of estimated yield PDF and (b) the proportion of $\mathcal{G_S}$ with different ρ_s.

and best corner (BC), to account for die-to-die (D2D) process variations. All the internal nodes that lie in a region spanned by the candidate inputs and outputs are put into a list $\mathcal{G_S}$ and $\mathcal{G_H}$ (L11 and L12) for subsequent simulation with MC samples; the nodes not in this list are completely ignored.

B. Experiments

The algorithm *Node_Filtering* and subsequent simulation to obtain timing yield were implemented in SIS [5]. We performed experiments on a set of 14 sequential circuits taken from ISCAS and open cores [6]. They were synthesized with SIS and mapped into a 45-nm gate library, which we built with a predictive model [7]. The threshold voltage (V_t) was chosen as a source of WID, as well as D2D, process variation. We assumed that the threshold voltage of an nMOS device at NC would have a normal distribution with a mean (μ_w) of 200 mV and a standard deviation (σ_w) of 20 mV, with a V_{dd} of 1.0 V; a pMOS device is assumed to follow the same normal distribution with a mean of -200 mV. To model the pin-to-pin delay of each gate, a SPICE simulation was performed for seven different values of V_t ($\mu_w + k\sigma_w$, where $k = -3, -2, \ldots, 2, 3$).

To assess the effectiveness of *Node_Filtering*, we used 1% for ρ_s and 5% for ρ_h, which seemed to be reasonable values under the trade-off between the error of simulation and the proportion of nodes that are selected. This is based on Fig. 2, where we show the error of standard deviation of estimated yield PDF from N_{che} samples with *Node_Filtering* (compared to the yield PDF from N_{che} samples without *Node_Filtering*) and the proportion of $\mathcal{G_S}$ while we change the value of ρ_s. The number of gates that belong to $\mathcal{G_S}$ was 9% on average; corresponding number for $\mathcal{G_H}$ was 4%. More flip-flops tend to be left out under hold time constraint because there are, in general, more short paths whose delay is comparable to the shortest delay than long paths whose delay is comparable to the longest delay.

To assess the accuracy of yield estimate, we compared three different methods: N_{che} number of MC samples with simulating all circuit nodes, which serve as a reference; N_{new} samples with simulating all circuit nodes (denoted as RS, for reduced sample); and N_{new} samples with *Node_Filtering* (denoted as RS+NF). The percentage error of yield estimate from RS and RS+NF were checked for $y = 0.90$ and $y = 0.95$. We tried two different confidence levels, 0.95 and 0.99; ε was fixed to 0.01. If we compare RS and RS+NF methods for the same confidence level and the same target yield, the latter has more errors, as it must; however, for all circuits we tried, the error always remained within $\pm 1\%$, thus satisfies ε. The error for $y = 0.95$ was smaller than that for $y = 0.90$, which is understandable because estimating higher yield is easier since yield becomes more deterministic.

We also measured the runtime of RS+NF method for four different combinations of confidence level and target yield: the speedup was 70 ($c = 0.95$ and $y = 0.90$), 241 ($c = 0.99$ and $y = 0.90$), 27 ($c = 0.95$ and $y = 0.95$), and 125 ($c = 0.99$ and $y = 0.95$) on average of all circuits. Note that there is more speedup when confidence level is higher, which is because of increasing difference of N_{che} and N_{new} with increasing confidence level (see Table I). Therefore, when confidence level is low, the speedup is mainly contributed by simulation of selected nodes, but when confidence level is high, it is reduced number of samples N_{new} that mostly contribute to the speedup.

IV. Conclusion

We have developed a closed-form expression that produces a sample number much closer (3× to 8×) to the minimum than the number provided by Chebyshev's inequality (5× to 15×). Since the sample number from new expression is still too large compared to the minimum, a further work needs to be done to refine the expression.

We have also proposed a simple heuristic algorithm to select the nodes that are likely to affect timing yield. Only 9% of gates under max-delay constraint and 4% under min-delay constraint are left, which greatly helps reduce simulation time. Applying the two proposed methods together reduces the runtime of MC method by 27× to 125×.

References

[1] P. Zuchowski, P. Habitz, J. Hayes, and J. Oppold, "Process and environmental variation impacts on ASIC timing," in *Proc. Int. Conf. on Computer Aided Design*, Nov. 2004, pp. 336–342.

[2] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, "Statistical timing analysis: from basic principles to state of the art," *IEEE Trans. on Computer-Aided Design*, vol. 27, no. 4, pp. 589–607, Apr. 2008.

[3] S. Naidu, "Timing yield calculation using an impulse-train approach," in *Proc. Asia South Pacific Design Automation Conf.*, Jan. 2002, pp. 219–224.

[4] L. Scheffer, "The count of Monte Carlo," in *ACM/IEEE Int. Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, Feb. 2004.

[5] E. Sentovich *et al.*, "SIS: a system for sequential circuit synthesis," May 1992, Tech. Rep. UCB/ERL M92/41.

[6] OpenCores. [Online]. Available: http://www.opencores.org/

[7] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45nm design exploration," in *Proc. Int. Symp. on Quality Electronic Design*, Mar. 2006, pp. 585–590.