INTEGRATION, the VLSI journal 48 (2015) 101–108

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

Stochastic logical effort as a variation aware delay model to estimate timing yield ☆

Alp Arslan Bayrakci

Department of Computer Engineering, Gebze Institute of Technology, Turkey

ARTICLE INFO

Article history:
Received 18 July 2013
Received in revised form
27 July 2014
Accepted 29 July 2014
Available online 8 August 2014

Keywords:
Logical effort
Timing yield estimation
Statistical timing analysis
Monte Carlo

ABSTRACT

Considerable effort has been expended in the EDA community during the past decade in trying to cope with the so-called *statistical timing* problem. In this paper, we not only present a fast and approximate gate delay model called stochastic logical effort (SLE) to capture the effect of statistical parameter variations on the delay but also combine this model with a previously proposed transistor level smart Monte Carlo method to construct ISLE timing yield estimator. The results demonstrate that our approximate SLE model can capture the delay variations and ISLE achieves the same accuracy as the standard Monte Carlo estimator with a cost reduction of about 180 × on the average for ISCAS'85 benchmark circuits and in the existence of both inter- and intra-die variations.

1. Introduction

Decreasing sizes of transistors result in manufacturing of digital integrated circuits (IC) to become much more difficult and prone to variations of parameters like transistor gate length, threshold voltage, etc. Performance (speed) variability due to the statistical parameter variations and environmental fluctuations has become more significant. IC designers need to estimate the timing yield and optimize their design accordingly until it reaches the desired yield before manufacturing. A survey [1] in 2011 by Solido Design Automation over 486 IC design professionals shows that two-thirds of designers and managers name variation-aware design as a top segment where technology advancement is needed.

In traditional VLSI design methodologies, designers prefer using Spice tool for detailed transistor level (TL) circuit simulations as a final verification before timing sign-off because of its accuracy. One would ideally like to perform a similar transistor-level, but statistical timing analysis for timing yield estimation. According to the above survey, more than half of the participant IC designers and managers want Spice simulators to be variation aware in the first place. Taiwan Semiconductor Manufacturing Company (TSMC) has already announced the insertion of transistor-level statistical timing analysis into its reference design flow in order to enhance timing accuracy [2]. There has been intense academic research on statistical timing analysis and timing yield estimation topics especially in the last decade [3,4]. The researchers have to cope with hard problems like modeling inter- and intra-die variations with spatial correlations, accurate delay approximations without solving the actual non-linear and differential delay equations, propagation of the non-Gaussian random variables, etc. Previously negligible problems have become more and more important with the shrinking technology. Today, intra-die variations are at least as important as inter-die variations [5]. The combination of all these problems either increases the computational complexity of the solution or decreases the accuracy of the resultant timing yield estimate. Making too many assumptions to decrease complexity results in far-off estimates. The most accurate approach in statistical timing is the Monte Carlo (MC) method based on costly transistor level Spice simulations, which is the called golden method, however it is computationally too complex to be applicable.

The main contributions of this paper can be summarized as follows: a variation aware delay model, stochastic logical effort (SLE), is proposed to capture the tendency of a gate's delay with respect to the random device parameters. The characterization of a standard cell library to prepare it for SLE delay computation and efficient methods for the computation of SLE model parameters are presented. SLE is combined with a previously proposed importance sampling based timing yield estimation technique [6] to build a new estimator called ISLE. The theoretical error of the resultant ISLE estimator is derived in detail. The empirical tests over ISCAS'85 benchmark circuits considering both inter- and intra-die variations with spatial correlations show the accuracy of the SLE method. The results show that ISLE timing yield estimation is about 180 times faster on the average than the traditional methods with the same accuracy. ISLE is not meant to

☆This research is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under the Project number 112E237.

http://dx.doi.org/10.1016/j.vlsi.2014.07.003
0167-9260/

be a replacement of the less accurate but faster statistical static timing analysis methods, but instead a complementary method to be used as a final verification for the statistically critical paths in the circuit.

In Section 2, we present the variation aware gate delay model SLE. In Section 3, ISLE yield estimator is explained and a precise theoretical error analysis is given. Finally in Section 4, we present the experimental results and the computational cost.

2. Variation aware gate delay model: stochastic logical effort (SLE)

Section 2.1 provides an overview of the well-known logical effort approach. In Section 2.2, we introduce SLE for approximating circuit delay in the presence of statistical variations. Section 2.3 explains the extraction of the model parameters for the SLE gate delay models, Section 2.4 presents the characterization of a cell library and Section 2.5 explains the detection of the SLE parameter values for a given sample point from the random parameter space.

2.1. Logical effort

The logical effort formalism [7] is a fast and efficient way of determining the delay of a path in a digital circuit. The path delay is simply the sum of the delays of the gates on the path, and the delay of a logic gate r is approximated as

$$d_r^{LE} = \tau d \tag{1}$$

where d_r^{LE} is the absolute delay of a gate measured in seconds, τ is the delay of a parasitic-capacitance-free *reference inverter* driving another identical inverter and d is the delay of the logic gate expressed in units of τ. The d factor in (1) models the gate delay and is given by

$$d = (p + gh) \tag{2}$$

where p represents the intrinsic (parasitic) delay, g is the logical effort and h is the electrical effort or electrical fanout. Logical effort g is a measure of the complexity of a gate. It depends only on the gate's topology and is independent of the size and the loading of the gate. Parasitic delay p expresses the intrinsic delay of the gate due to its own internal parasitic capacitance, and it is largely independent of the sizes of the transistors in the gate. The fanout h is the ratio of the load capacitance of the logic gate to the capacitance of a particular input.

2.2. From logical effort (LE) to stochastic logical effort (SLE)

Eqs. (1) and (2) provide a way of decomposing the effects of statistical parameter variations on gate delays. In a different context, Sutherland et al. [7] analyzed different semiconductor processes with varying supply voltages, and observed that almost all the effect of process parameters and supply voltage on gate delay is captured by the reference inverter delay (τ in (1)), even when the parameters vary over a large range spanning different fabrication processes. The logical effort g and the unitless parasitic delay p of a gate exhibit relatively little variation with process parameters and supply voltage. Exploiting this observation in the context of timing yield analysis in [8] a stochastic logical effort (SLE) model was proposed where the delay of a gate was modeled as

$$d_r^{LE}(X) = \tau(X)(p + gh) \tag{3}$$

where X is a vector of random variables, each component of which represents a different statistical circuit or process parameter and $\tau(X)$ is the reference inverter delay when the parameters are given by X. As is apparent in this equation, in the stochastic logical effort approximation, all process and environmental variations are captured by the statistical variable τ while g, p and therefore d are assumed to be independent of process parameters. We refer to the approximation given in (3) as *first-degree stochastic logical effort* (abbreviated as *SLE.d1*).

In this paper, we introduce a further refinement of this approximation described by the following equation:

$$d_r^{LE}(X) = \tau(X)(p(X) + g(X)h) \tag{4}$$

where the dependency of p and g on X is also modeled. We call this model *second-degree stochastic logical effort* (*SLE.d2*). As will become apparent later in the paper, *SLE.d2* is much more accurate but computationally more expensive for the first characterization phase.

In both versions of SLE, in order to compute the delay of a path π in a circuit, we simply add the delays of the gates on π:

$$d_\pi^{LE}(X) = \sum_{r=1}^{k} d_r^{LE}(X) \tag{5}$$

Here $d_r^{LE}(X)$ is the delay of the rth gate on the path π. $d_r^{LE}(X)$ is computed by evaluating (3) for *SLE.d1* and (4) for *SLE.d2*. For this evaluation, a full transistor-level simulation of the whole circuit containing the logic path is not necessary. However, the values of $\tau(X)$ (for both *SLE.d1* and for *SLE.d2*) and $p(X)$ and $g(X)$ (for *SLE.d2*) at a given X are needed. For this purpose, we construct look-up tables of the SLE parameters $\tau(X)$, $p(X)$ and $g(X)$ as the pre-characterization of a standard cell library. In this case, no circuit simulations will be needed when evaluating the SLE delay formulas for circuits that are built using gates from such a pre-characterized library. The extraction of the SLE model parameters and the pre-characterization of a standard cell library are presented in the following sections.

2.3. Extraction of SLE model parameters

SLE parameters can be computed at a given X by running transistor level circuit simulations on small test circuits which contain only the reference inverter (for $\tau(X)$) or the gate under consideration (for $p(X)$ and $g(X)$) together with a proper driver and load circuitry. Fig. 1 shows the test circuit constructed with only reference inverters to compute $\tau(X)$. For computing $\tau(X)$, first, the random parameters of all inverters in the figure are set according to X and then the number of inverters connected to the node 4 is iterated from 1 to 8. This means that h (fanout) is iterated from 1 to 8 as h is equal to the output load capacitance over input capacitance. Similarly input slope can be changed by changing the number of inverters connected to node 3 to collect results for different slopes. At each iteration, by performing TL transient

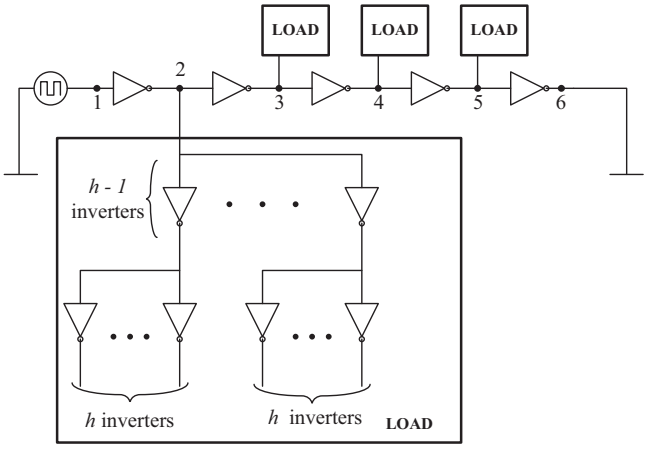

Fig. 1. Reference circuit for $\tau(X)$ computation.

analysis (SPICE simulations), the delay of the inverter between nodes 3 and 4 is recorded. As a result, for a sample X, a plot similar to Fig. 2 is obtained.

The x-axis in the plot is fanout, h, and the recorded delays are marked by the crosses. A line is fitted to the marked crosses as shown in the figure. This line has the slope $g.\tau(X)$. As g is 1 for the reference inverter by definition, the slope is equal to $\tau(X)$. Also the point where the line intersects the y-axis is equal to $p.\tau(X)$. The fairness of linear fanout–delay relationship assumption in the SLE formalism can be observed in the figure.

If *SLE.d2* is used, a similar extra test circuit with a similar plot should be used for each gate type[1] to compute $p(X)$ and $g(X)$. But this time, instead of inverters, the test circuit in Fig. 1 is constructed with the gate type, whose $p(X)$ and $g(X)$ values are computed. Then, the slope of the fitted line is used to compute $g(X)$, whereas the point of intersection with the y-axis is used to compute $p(X)$ of the corresponding gate type. For the more accurate *SLE.d2* gate delay model, the construction of look-up tables for $p(X)$ and $g(X)$ is required in addition to $\tau(X)$ table, which are constructed only once during the characterization of a standard cell library.

2.4. Characterization of a cell library

When both inter- and intra-die variations are considered, SLE has to be able to compute the delay for any gate having any random parameter values. For this purpose, a table of $\tau(X)$ values for *SLE.d1* and also tables of $p(X)$ and $g(X)$ values for *SLE.d2* are required. In order to construct such tables, we first divide the random parameter space with equal intervals and then perform the transient Spice analysis as explained in Section 2.3 for each sample point separated with equal intervals.

In this paper, we assume two random parameters: gate length (L) and threshold voltage (V_t). Therefore, we divided the 2D random parameter space as shown in Fig. 3, but instead of 153×153 points in the figure, we used 20×20 sample points in order to perform less number of Spice TL simulations for the characterization of the standard cell library. But the resolution of the resultant tables is low as only parameter values corresponding to 20×20 samples are computed. We apply linear interpolation to increase the resolution of the look-up tables up to 153×153 as shown in Fig. 3.

2.5. Detection of the SLE parameters for any sample point X

After the characterization is over, in order to compute the delay of a gate, one has to draw the SLE parameters from the look-up tables for a given X. This is done by selecting the table entry that corresponds to the closest sample point in the random parameter space to the desired X. But if the closest point was computed by comparing the euclidian distance to each of the 153×153 sample points, this would consume too much time and affect the performance of SLE very bad, because this operation is performed for each gate and sample point during the run-time. Instead of that, we do the following: assume that we want to find the closest point to the given sample point shown in Fig. 3, whose corresponding gate length is A and threshold voltage is B. The closest point is (L_M, V_N), where M and N are computed as

$$M = Round\left(\frac{A - L_1}{\alpha}\right) + 1 \tag{6}$$

$$N = Round\left(\frac{B - V_1}{\beta}\right) + 1 \tag{7}$$

[1] Gate types refer to different gates with different numbers of inputs and different functionalities like AND, OR, NOR, NAND, etc.

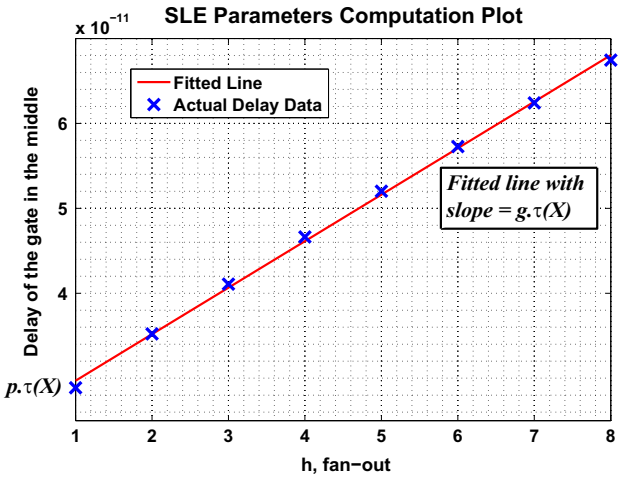

Fig. 2. Actual actual fanout vs. delay plot of the inverter for a sample point X.

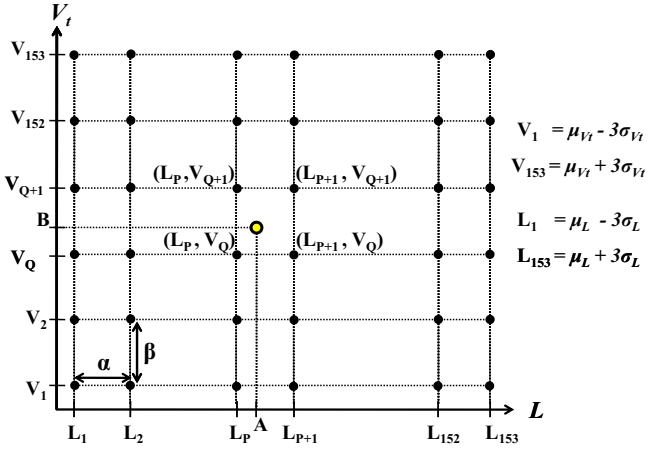

Fig. 3. The random parameter space divided by equal intervals and the detection of a closest point for a given $\mathcal{X}$. μ is the mean and σ is the standard deviation.

The required SLE parameters corresponding to the point (L_M, V_N) can now be used in the computation of the gate delay by (3) for *SLE.d1* and by (4) for *SLE.d2*.

3. Timing yield estimation based on SLE gate delay model and importance sampling

3.1. Preliminaries and previous work

Timing yield is the fraction of dies, which satisfy the timing requirements, in other words which have circuit delay smaller than a timing constraint T_c.

In our previous work [6], we had introduced a novel timing yield estimation methodology based on importance sampling. In this paper, we will utilize the same methodology, but this time in conjunction with the SLE gate delay model. The timing yield estimation model is based on the transistor level Monte Carlo estimation of timing yield (*Yield*) or loss (*Loss*) where

$$Loss = 1 - Yield = \int I(T_c, X) f(X)\, dX \tag{8}$$

as the mean of the indicator random variable $I(T_c, X)$ over the PDF $f(X)$, where $f(X)$ is the joint probability density function for the random parameters inside the circuit like gate length and

threshold voltage. The indicator variable above is expressed as

$$I(T_c,X)=\begin{cases}1 & \text{if} \quad d_C(X)>T_c\\ 0 & \text{if} \quad d_C(X)\le T_c\end{cases} \tag{9}$$

and $d_C(X)$, which is the circuit delay corresponding to X is expressed as

$$d_C(X)=\max_{\pi\in\Pi_{crit}} d_\pi(X) \tag{10}$$

A path π in a circuit $\mathcal{C}$ is a sequence of gates $g_0,g_1,g_2,\ldots,g_n$ where g_0's inputs are primary inputs of the circuit and g_n's output is a primary output of the circuit. We denote by Π_{crit} the set of statistically critical paths. $d_\pi(X)$ in (10) represents the delay of the critical path π and is computed using precise transistor level Spice simulations when the random variables inside the circuit are set to X.

The corresponding standard Monte Carlo estimator for *Loss* in (8) is expressed as

$$Loss_N=(1/N)\sum_{i=1}^{N} I(T_c,X_i) \tag{11}$$

where X_i's are the drawn samples according to $f(X)$ and T_c is the timing constraint. With the MC method, full circuit simulations (TL Spice simulations of the whole circuit containing the paths under consideration) must be performed for each sample point, X_i, in order to compute $d_C(X_i)$ and determine whether $I(T_c,X_i)=1$ or 0. The MC method is widely used as a golden reference in the literature in assessing the accuracy and efficiency of timing yield estimation techniques. However, it is generally believed that it cannot be used in practice for estimating timing yield as it requires too many costly full circuit simulations for acceptable accuracy, even though there are some arguments to the contrary [9]. In the rest of this paper, the loss estimator in (11) is referred to as the standard MC (STD-MC) estimator. From the central limit theorem, the probability that the loss estimates of the STD-MC estimator will be in a range $\pm 1.96(\sigma/\sqrt{N})$ around the actual loss (*Loss*) where

$$\sigma^2=\int_\Omega I(T_c,X)f(X)\,dX-Loss^2 \tag{12}$$

is written as

$$P\left(Loss-1.96\frac{\sigma}{\sqrt{N}}\le Loss_N\le Loss+1.96\frac{\sigma}{\sqrt{N}}\right)=0.95 \tag{13}$$

Therefore, the error of the Monte Carlo estimators in the form of (11) with more than 95% confidence is expressed as

$$\left|Error\right|\approx\frac{2\sigma}{\sqrt{N}} \tag{14}$$

At that point, [6] proposes to use the importance sampling estimator to speed-up STD-MC, as shown below:

$$Loss_N^{IS}=\frac{1}{N}\sum_{i=1}^{N} I(T_c,X_i)\frac{f(X_i)}{\tilde{f}(X_i)} \tag{15}$$

which draws the samples X_i from another biasing distribution $\tilde{f}$. The choice of this biasing distribution is the key issue to speed up STD-MC. In this paper, we use SLE gate delay model to determine a biasing distribution for the IS estimator in (15).

3.2. Importance sampling with stochastic logical effort (ISLE)

Loss can also be estimated based on the SLE formalism, without performing any full circuit simulations. The delay of a circuit can be computed analytically based on the SLE formalism as follows:

$$d_C^{LE}(X)=\max_{\pi\in\Pi_{crit}} d_\pi^{LE}(X) \tag{16}$$

where $d_\pi^{LE}(X)$ is evaluated using the SLE formula in (5) and using *SLE.d1* or *SLE.d2*. We define a new indicator random variable $I^{LE}(T_c,X)$, which takes the value 1 if the delay of a circuit computed analytically using the SLE equations exceeds the target delay T_c, i.e., $I^{LE}(T_c,X_i)$ is 1 if $d_C^{LE}(X_i)>T_c$ and 0 otherwise. The loss estimator based on this new indicator variable takes the form

$$Loss_N^{LE}=\frac{1}{N}\sum_{i=1}^{N} I^{LE}(T_c,X_i) \tag{17}$$

In computing $Loss_N^{LE}$ above, no full circuit simulations are performed. Only simple evaluations of the SLE delay formulas are needed, based on the pre-characterizations of $\tau(X)$, $p(X)$ and $g(X)$. In contrast, the STD-MC loss estimator requires N full circuit TL simulations, one for every sample. The loss estimator in (17) will be referred to as the SLE-MC estimator in the rest of this paper.

The estimation of loss based on the STD-MC estimator is obviously much more accurate than the one based on the SLE-MC estimator, but much more costly. We use the cheap SLE-MC estimator not by itself for yield estimation, but by in a novel approach to construct a much faster IS-based loss estimator called ISLE. The proposed biasing distribution $\tilde{f}(X)$ to be used in ISLE is

$$\tilde{f}(X)=\frac{I^{LE}(T_c^\epsilon,X)f(X)}{Loss^{LE,\epsilon}} \tag{18}$$

where T_c^ϵ is $T_c-\epsilon$ and ϵ is an adaptive margin introduced in [6]. Substituting the biasing distribution $\tilde{f}$ in (18) and performing some simplifications based on the fact that $I^{LE}(T_c^\epsilon,X_i)$ takes the value 1 for all samples drawn from $\tilde{f}(X)$, we arrive at

$$Loss_N^{ISLE}=\frac{Loss^{LE,\epsilon}}{N}\sum_{i=1}^{N} I(T_c,X_i) \tag{19}$$

where the samples X_i are drawn from $\tilde{f}(X)$ instead of $f(X)$.

3.3. Quantifying variance reduction due to ISLE

The error of an estimator is the deviance of the estimator's result from the actual loss as explained in Section 3.1 for a general estimator. In this section, the errors of the STD-MC and ISLE estimators are derived and the results are compared. The paper [6], which proposes the utilization of importance sampling, also uses an identical theoretical error notation, however, it does not show the derivation of the proofs, which we show here.

Theorem 3.1. *The error of the STD-MC estimator in (11) obtained with N full-circuit simulations is*

$$Error_{MC}=2\sqrt{Loss.Yield}/\sqrt{N} \tag{20}$$

with more than 95% confidence.

Proof. By (14), the error of the STD-MC estimator for loss using N full-circuit simulations is $2\sigma/\sqrt{N}$ where σ^2 is the variance of the indicator random variable $I(T_c,X)$ with PDF $f(X)$. The mean of $I(T_c,X)$ is equal to the actual timing loss. σ^2 is computed as

$$\sigma^2=\int_\Omega I(T_c,X)^2f(X)\,dX-Loss^2 \tag{21}$$

$I(T_c,X)$ is either 1 or 0, thus, $I(T_c,X)=I(T_c,X)^2$. Eq. (21) becomes

$$\sigma^2=Loss-Loss^2=Loss(1-Loss)=Loss.Yield \tag{22}$$

The error of the STD-MC estimator is thus given by (20). □

Theorem 3.2. *The error of the ISLE estimator in (19) when N full circuit simulations are performed is*

$$Error_{ISLE}=2\sqrt{Loss.(Loss^{LE,\epsilon}-Loss)}/\sqrt{N} \tag{23}$$

with more than 95% confidence.

Proof. By (14), the error of the ISLE estimator for loss using N full-circuit simulations is $2\tilde{\sigma}/\sqrt{N}$ where $\tilde{\sigma}^2$ is the variance of the random

variable $I(T_c,X)f(X)/\tilde{f}(X)$ with PDF $\tilde{f}(X)$. The mean of this random variable is equal to the actual timing loss. $\tilde{\sigma}^2$ is computed as

$$\tilde{\sigma}^2 = \int_{\Omega} \left(\frac{I(T_c,X)f(X)}{\tilde{f}(X)} \right)^2 \tilde{f}(X)\, dX - Loss^2 \tag{24}$$

Substituting $\tilde{f}(X)$ from (18) and using the fact that $I(T_c,X)^2 = I(T_c,X)$ we obtain

$$\tilde{\sigma}^2 = \int_{\theta} \frac{I(T_c,X)f^2(X)}{\frac{I^{LE}(T_c^{\epsilon},X)f(X)}{Loss^{LE,\epsilon}}} dX - Loss^2 \tag{25}$$

θ denotes the subregion of Ω in which $\tilde{f}(X)$ is non-zero. From (18), $\tilde{f}(X)$ is zero when $I^{LE}(T_c^{\epsilon},X)$ is zero (and thus $I(T_c,X)=0$, if the margin ϵ is chosen properly). When $\tilde{f}(X)$ is non-zero, $I^{LE}(T_c^{\epsilon},X)=1$. Thus

$$\tilde{\sigma}^2 = Loss^{LE,\epsilon} \int_{\theta} I(T_c,X)f(X)\, dX - Loss^2 = Loss.(Loss^{LE,\epsilon} - Loss) \tag{26}$$

The error of the ISLE estimator is thus given by (23). □

If the same number of full circuit simulations N is used with both methods, then the ratio of the errors of the estimators is given by

$$Error\ Ratio = \frac{Error_{MC}}{Error_{ISLE}} = \sqrt{\frac{Yield}{(Loss^{LE,\epsilon} - Loss)}} \tag{27}$$

Alternatively, suppose a bound on the allowable estimation error is given. The ratio of the minimum number of full circuit simulations required by the two approaches to achieve this same error bound is given by

$$\texttt{Speedup} = \frac{N_{MC}}{N_{ISLE}} = \frac{Yield}{(Loss^{LE,\epsilon} - Loss)} = Error\ Ratio^2 \tag{28}$$

As is apparent from (27) and (28), while $Loss^{LE,\epsilon}$ approaches the real loss $Loss$, the improvement that ISLE offers over STD-MC increases. If Speedup in (28) is large, one might conclude that TL simulations are not needed and $Loss^{LE,\epsilon}$ can simply be used as an accurate loss estimate. However, this conclusion is not correct. $Loss^{LE,\epsilon}$ is computed using (17), where $T_c^{\epsilon} = T_c - \epsilon$ with ϵ as the margin parameter, which is determined adaptively by the help of the TL simulations [6].

4. Results

4.1. Experimental setup

Among all process parameters, the most significant are channel length and threshold voltage [10]. Therefore, in our experiments, we assumed two random device parameters: transistor gate length (L) and threshold voltage (V_t), whose statistical variations are regularly reported by International Technology Roadmap for Semiconductors (ITRS) reports due to their impact on the performance of the integrated circuits. The random parameter variations are set according to the 2011 report of ITRS [11]:

- Effective channel length L with a $3\sigma/\mu$ ratio of 12%.
- Threshold voltage V_t with a $3\sigma/\mu$ ratio of 20%.

We have considered both inter- and intra-die variations as both share a similar portion of the statistical process variations [5,12]. Half of the variation is assumed to come from inter-die and the other half from intra-die variations. An important property of the intra die variations that must be modeled to catch the reality is the correlation of the statistical device parameters that increases while the distance in between the gates decreases, i.e. spatial correlation. In order to take into account the spatial correlations, a 4-level version of the quad-tree model proposed in [13] is constructed. According to this model and for each random parameter, we employ 85 independent random variables to model spatial correlations. In our case, 170 independent random variables are employed for two random parameters.

ISCAS'85 benchmark circuits [14] are utilized in the experiments. Three statistically critical, statically sensitizable (true) paths are extracted representing the statistically critical paths set (Π_{crit}) for each test circuit. They include 22 different types of logic gates with an average path length of 30 gates. The circuits are realized by NanGate 45 nm Open Standard Cell Library [15].

NgSpice [16] open source Spice simulator is used to collect precise transistor level Spice simulation timing data. We modified the source code of the NgSpice simulator so that 45 nm transistor model file [17] data is set as default and also both the gate length and threshold voltage values for each transistor in the circuit can be set independently inside the Spice circuit deck. A total of about 1.5 million transistor level simulations are performed in parallel on a system with two Xeon E5-2620 6 core 2 GHz processors in order to collect the exact delays and compute the actual loss values for the benchmark circuits. The simulations have prolonged more than 1 month.

4.2. Empirical accuracy of SLE gate delay model

We use SLE as an on the back of an envelope method to approximate the change in the delay of a gate when the statistical parameters of the gate differ. In this section, we will analyze the accuracy of SLE model with respect to the golden TL circuit simulations. The delay of each benchmark circuit is computed for 50,000 different sample points drawn from $f(X)$, where $f(X)$ is the joint distribution of L and V_t according to the variation model explained in the previous section. As a result, the circuit delay is computed for each sample point, for each test circuit and by both TL circuit (Spice) simulation and the stochastic logical effort gate delay models (*SLE.d1* and *SLE.d2*). Table 1 demonstrates the accuracy of SLE methods with respect to Spice TL simulation.

The first column in Table 1 is the test circuit, the second and third columns show the normalized root mean square (NRMSE) error when *SLE.d1* and *SLE.d2* are used respectively for the circuit delay approximation. The root mean square error (RMSE) is computed as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (d_C(X_i) - d_C^{LE}(X_i))^2}{N}} \tag{29}$$

where $d_C(X_i)$ is the actual circuit delay computed by Spice TL simulation for the given X_i sample point and $d_C^{LE}(X_i)$ is the approximate circuit delay computed by the SLE delay model (either by *SLE.d1* or *SLE.d2*) for the same X_i. N is the number of drawn sample points, which is 50,000 in this case. The NRMSE can

Table 1
The accuracy of *SLE.d1* and *SLE.d2*.

Test circuit	NRMSE (%)	
	SLE.d1	*SLE.d2*
c432	3.19	5.09
c499	4.78	1.01
c880	3.55	0.39
c1355	2.92	2.01
c1908	4.46	0.43
c2670	3.40	0.54
c3540	3.72	0.30
c5315	3.61	0.26
c7552	3.06	0.64

be computed through dividing the RMSE by the sample mean of the actual circuit delay. It can be deduced from Table 1 that the accuracy of *SLE.d2* is much better than *SLE.d1* as expected. However, for c432, *SLE.d2* delay has a more shifted mean delay value which results in a greater NRMSE error. The reason for that is the exceptionally unbalanced gate loads in c432 circuit. Nonetheless, if we equalize the means of *SLE.d1* and *SLE.d2* delays with the mean of the actual delays, the resultant NRMSE errors for c432 would become as 2.57% and 0.64% respectively, which shows the fact that even for c432 *SLE.d2* performs much better than *SLE.d1* in terms of catching the variations.

To gauge the accuracy of *SLE.d1* and *SLE.d2*, the scatter plot in Fig. 4 shows the delays computed by SLE formulas versus the actual delays computed by TL simulations for each of the 50,000 sample points. The delay computed by *SLE.d1* versus the actual delay (circle) and the delay computed by *SLE.d2* versus the actual delay (cross) are shown in the plot. As seen in this plot, SLE methods have a tendency similar to the actual delays. As expected, *SLE.d2* is more accurate, which comes at the cost of having statistical pre-characterizations for parasitic delay p and logical effort g for all the gates in the library, in addition to the reference inverter delay τ.

4.3. Empirical accuracy of ISLE loss estimation

With the results that we present in this section, we compare the accuracy and the efficiency of our improved ISLE estimator in (19) against the standard Monte Carlo (STD-MC) estimator in (11). In order to empirically measure the error in the loss estimates obtained by the STD-MC estimator and ISLE estimator, we perform independent repetitions of the same experiment run. In doing so, we empirically compute the error (or variance) achieved by both the loss estimators.

We perform 250 independent repetitions of the same experiment (evaluation of the estimator). These 250 independent runs constitute the samples of the loss estimator and the variance and error of the loss estimator is computed over these 250 loss estimations. In each independent run, 200 samples are drawn independently from the joint PDF $f(X)$ in the parameter space. For the ISLE estimator, most of the 200 samples are discarded as explained in Section 3.2 based on the evaluation of the *SLE* equations and a reduced number (N_{ISLE} on the average, showing negligible variation from run to run) of TL simulations are performed. N_{ISLE} includes all the TL circuit simulations required to compute Loss_N^{ISLE} in (19). In evaluating the STD-MC estimator, we draw $N_{TL} = N_{ISLE}$ samples randomly. For the STD-MC estimator, the results of TL circuit simulations performed at every one of the N_{TL} sample points are used. By this setup, we utilize the same number of TL simulations at each estimation of both ISLE and STD-MC loss estimators for a fair comparison. We collect the resultant 250 loss estimates coming from STD-MC estimator and 250 estimates from ISLE estimator, where both estimators use the same number of TL simulations for each estimate. We compute empirical `Speedup` in (28) using (27), i.e. by dividing the variance of STD-MC estimator to the variance of ISLE estimator over these 250 estimates and around the actual loss.[2] The actual loss value is computed using STD-MC over all the $50{,}000 = 250 \times 200$ sample points generated during all of the 250 runs.

The $Loss^{LE,\epsilon}$ value that is needed for computing the IS estimator in (19) is computed using the SLE based MC estimator (SLE-MC) in (17) using all the 50,000 sample points. The computation of it brings a negligible cost as it is based on very simple and fast SLE gate delay models and does not require any TL simulation. This will be clarified in the next section.

The `Speedup` that we report for the ISLE estimator over the STD-MC estimator can be interpreted in two ways: it represents the ratio of the number of TL circuit simulations required by the STD-MC and ISLE estimators to achieve the same accuracy (error), as given by (28). Secondly, `Speedup` shows how much smaller the variance of ISLE estimator is than the STD-MC estimator although they both perform the same number of TL simulations.

The timing constraint values are set for a loss of about 10%. In order to detect the `Speedup`, we compute three loss estimates using three estimators for each test circuit in ISCAS'85 benchmark:

- the standard MC (STD-MC) estimator,
- the ISLE estimator based on *SLE.d1*,
- the ISLE estimator based on *SLE.d2*.

The `Speedup` is computed for two different ISLE versions, one based on *SLE.d1* and the other based on *SLE.d2* by dividing the variance of STD-MC estimates to the variance of ISLE estimates around the actual *Loss* and with the same number of TL simulations. The actual *Loss* is computed using STD-MC with 50,000 sample points, i.e. 50,000 TL circuit simulations. The results in Table 2 show that both versions of our ISLE yield estimator achieve cost reduction over the STD-MC estimator for the same accuracy. In other words, both versions of ISLE estimator have better accuracies than STD-MC estimator when the same number of TL simulations are utilized. ISLE based on *SLE.d2* performs much better, achieving two orders of magnitude cost reduction in the worst-case, whereas the cost reduction achieved by ISLE based on *SLE.d1* goes down to 8 in the worst-case. The average `Speedup` for *SLE.d1* is 12 whereas for *SLE.d2*, it is 179.

In order to visualize the `Speedup` or variance reduction gathered by ISLE based on *SLE.d2* over STD-MC, it is good to plot loss estimates of both estimators, where both of them use the same number of TL simulations for loss estimation, i.e. they have the same computational cost. Fig. 5 demonstrates these loss estimates for each of the 250 sets for c1908 test circuit as an example. This plot clearly shows that every estimate of ISLE based

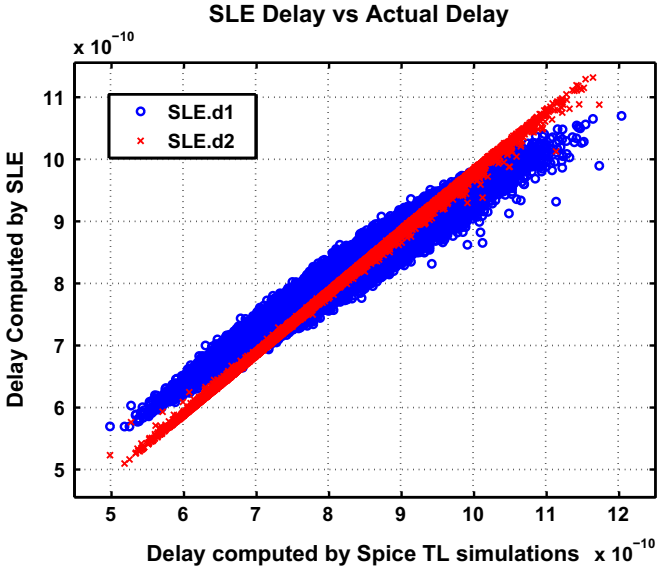

Fig. 4. Scatter plot for test circuit c1355 (the plots are similar for all benchmark circuits).

[2] Combining (22) and (26) with (28), it can be seen that `Speedup` is equal to the ratio of the variances for the loss estimates obtained by the two estimators with the same number of samples (TL simulations).

Table 2
The resultant Speedup over STD-MC when *SLE.d1* or *SLE.d2* based ISLE estimator is used for loss estimation.

Test circuit	*Loss* (%)	Speedup	
		SLE.d1	*SLE.d2*
c432	10.91	11	130
c499	10.93	8	103
c880	10.64	12	157
c1355	10.47	13	232
c1908	10.09	13	123
c2670	10.09	11	210
c3540	10.17	11	195
c5315	10.76	14	237
c7552	10.63	12	225

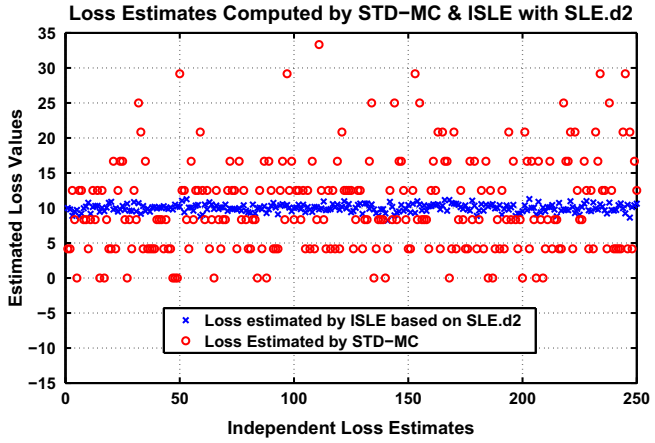

Fig. 5. The 250 independent loss estimates computed by STD-MC and ISLE based on *SLE.d2* for Test Circuit c1908. Both estimators use the same number of TL simulations for each estimate.

on *SLE.d2* is much closer to the actual *Loss*, which is 10.09% for c1908 as shown in Table 2.

4.4. Analysis of computational cost

The TL Spice simulation requires the numerical solutions of many non-linear, differential equations. It is a very precise, but computationally too complex method. ISLE can reduce the number of TL simulations without decreasing the accuracy. Yet, it comes with two additional costs: one-time pre-characterization of a standard cell library and the computation of $Loss^{LE,\epsilon}$. In our implementation, we required about 1.5 million TL Spice simulations of the simple test circuit in Fig. 1 for the pre-characterization of NanGate's 45 nm Open Cell Library. This is a one-time cost for a standard cell library that will never be repeated in the ISLE timing yield analysis of any circuit designed with the same cell library. Therefore, we do not take it into account in the complexity analysis. Second overhead source, i.e. the computation of $Loss^{LE,\epsilon}$ based on (19), requires circuit delay computation for each sample point but using only linear SLE equations in (4) without requiring any TL simulations. This computation is much faster than the TL simulation and can be computed in parallel with the TL simulations required by ISLE, therefore we did not reserve an additional overhead for it in Speedup computations.

As a sample case, we computed the empirical cost results for a sample path with 25 gates from c880 on our Xeon 2 GHz 6 core machine. Spice TL simulation to compute the delay of this path for one sample point requires about 12 s, whereas the computation of the path's delay using SLE formalism for one sample point requires 3.84 ms. In ISLE, the additional overhead is due to the computation of the $Loss^{LE,\epsilon}$, which requires the computation of path delay using SLE for all sample points. Looking at Table 2, the Speedup for c880 is 157, therefore assuming 50,000 sample points as we did in this paper, the computational cost of STD-MC and ISLE with the same accuracy can be computed as

$$Cost^{STD-MC} = 50,000 \times 12\ \mathrm{s} = 166.7\ \mathrm{h}$$

$$Cost^{ISLE} = \frac{50,000 \times 12\ \mathrm{s}}{157} + 50,000 \times 3.84\ \mathrm{ms} = 1.1\ \mathrm{h}$$

As seen in this sample case the additional overhead ($50,000 \times 3.84$ ms) of ISLE is negligible and even if considered, it can only decrease the Speedup from 157 to 152. It constitutes only about 5% of $Cost^{ISLE}$. It can also be computed in parallel with the TL simulations resulting in almost zero overhead.

5. Conclusion

We have demonstrated in this paper that for a variance aware IC delay analysis *stochastic logical effort* can be used to approximately but efficiently capture the gate delay variations due to the parameter variations and that importance sampling in conjunction with stochastic logical effort (ISLE) can serve as a very accurate yet computationally viable timing yield estimation method as a final stage verification.

The proposed SLE formalism based ISLE timing yield estimation is applied to ISCAS'85 test circuits while both inter-die and intra-die variations with spatial correlations are taken into account. The results show that *SLE.d2* based ISLE can estimate timing yield on the average 179 times faster than the TL simulation based standard Monte Carlo although both estimates have the same accuracy level.

References

[1] ⟨http://www.solidodesign.com/files/variation-aware-custom-design-survey-2011.pdf⟩. Variation Aware Custom IC Design Report 2011, Amit Gupta, President and CEO of Solido Design Automation.
[2] Taiwan Semiconductor Manufacturing Company (TSMC) Reference Flow 9.0 Press Release. New TSMC Reference Flow 9.0 Supports 40 nm Process Technology. ⟨http://www.tsmc.com⟩.
[3] D. Blaauw, K. Chopra, A. Srivastava, L. Scheffer, Statistical timing analysis: from basic principles to state of the art, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 27 (April) (2008) 589–607.
[4] Cristiano Forzan, Davide Pandini, Statistical static timing analysis: a survey, Integr. VLSI J. 42 (June (3)) (2009) 409–435.
[5] Ji-Seong Doh, Dae-Wook Kim, Sang-Hoon Lee, Jong-Bae Lee, Young-Kwan Park, Moon-Hyun Yoo, Jeong-Taek Kong, A unified statistical model for inter-die and intra-die process variation, in: International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), (September) 2005, pp. 131–134.
[6] Alp Arslan Bayrakci, Alper Demir, Serdar Tasiran, Fast monte carlo estimation of timing yield with importance sampling and transistor-level circuit simulation, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 29 (September) (2010) 1328–1341.
[7] I. Sutherland, B. Sproull, D. Harris, Logical Effort: Designing Fast CMOS Circuits, Morgan Kaufmann, USA, 1999.
[8] A. Demir, S. Tasiran, Stochastic logical effort: designing for timing yield on the back of an envelope, in: ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU), (February) 2006.
[9] L. Scheffer, The count of Monte Carlo, in: ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU), February 2004.
[10] Dennis Sylvester, Kanak Agarwal, Saumil Shah, Invited paper: variability in nanometer cmos: impact, analysis, and minimization, Integr. VLSI J. 41 (May 3) (2008) 319–339.

[11] ⟨http://www.itrs.net/⟩ International Technology Roadmap for Semiconductors (ITRS) Report 2011 Edition.
[12] J. Cong, P. Gupta, J. Lee, Evaluating statistical power optimization, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 29 (November (11)) (2010) 1750–1762.
[13] A. Agarwal, D. Blaauw, V. Zolotov, Statistical timing analysis for intra-die process variations with spatial correlations, in: ACM/IEEE International Conference on Computer Aided Design (ICCAD), 2003.
[14] F. Brglez, H. Fujiwara, A neutral netlist of 10 combinational benchmark circuits, in: Proceedings of the IEEE International Symposium on Circuits and Systems, 1985, pp. 695–698.
[15] ⟨http://www.nangate.com/⟩ NanGate 45 nm Open Cell Library.
[16] ⟨http://ngspice.sourceforge.net/⟩ NgSpice Circuit Simulator.
[17] ⟨http://ptm.asu.edu/⟩ Predictive Technology Model, 45 nm Model File.

Alp Arslan Bayrakci received the B.S. degree in electrical and electronics engineering from Middle East Technical University, Ankara, Turkey, in 2004 and direct Ph.D. degree in computer engineering from Koc University, Istanbul, Turkey, in 2010. He has been with the Gebze Institute of Technology, Kocaeli, Turkey, since June 2011. His current research interests include statistical timing analysis, computer arithmetic, hardware trojans and computer-aided design methodologies.