Gradient Importance Sampling: an Efficient Statistical Extraction Methodology of High-Sigma SRAM Dynamic Characteristics

Thomas Haine1, Johan Segers2, Denis Flandre1, David Bol1
1ICTEAM Institute, 2Institute of Statistics, Biostatistics and Actuarial Sciences
Université catholique de Louvain, Louvain-la-Neuve, Belgium
Email: thomas.haine@uclouvain.be, david.bol@uclouvain.be

Abstract—The impact of within-die transistor variability has increased with CMOS technology scaling up to the point where it has emerged as a systematic problem for the designer. Estimating extremely low failure rate, i.e. "high-sigma" probabilities, by the conventional Monte Carlo (MC) approach requires millions of simulation runs, making it an impractical approach for circuit designers. To overcome this problem, alternative failure estimation methodologies, which require a smaller number of runs have been proposed.

In this paper, we propose a novel methodology called “gradient importance sampling” (GIS) for fast statistical extraction of high-sigma circuit characteristics. It is based on conventional Importance Sampling combined with a gradient-based approach to find the most probable failure point (MPFP). By applying GIS to extract SRAM dynamic characteristics in 28nm FDSOI CMOS, we show that the proposed methodology is straightforward, computationally efficient and the results are in line with those obtained via standard MC. To the best of our knowledge, the GIS results are the best in their class for low failure rate estimation.

Index Terms—Gradient, Importance sampling, Monte Carlo, Low failure rate probability, Variability, High-sigma

I. INTRODUCTION

For many decades, the number of transistors on a chip has doubled every two years according to Moore’s law. While the growth rate may not be accurate any more, this trend continues to be true. Still, even though there are many more transistors in a circuit the overall chip yield must be preserved for profitability. This means that the failure rate of an individual circuit element such as an SRAM bitcell must decrease at the same time. In addition, the influence of random within-die variations on the circuit performance is increasing with progressive technology scaling. For these reasons, variability is amongst the most important concern for circuit designers these days. Variability-aware circuit-design requires performing an estimation of low failure rates, which is conventionally performed with a huge number of MC simulation runs, wasting computing resources and time. For example, targeting a yield of 99.9% for a 32kB SRAM, the bitcell failure rate must be below $3.8 \times 10^{-9}$. Therefore, on average $3.8 \times 10^{9}$ simulations need to be done to observe a single failing bitcell. Depending on the accuracy and confidence interval required, the number of simulation runs should be multiplied by a few orders of magnitude. This is obviously not practical in reality.

Various techniques have been proposed to overcome these issues. They can be classified in three categories: Firstly, analytical methodology starts by building a simplified and sometimes less accurate model of the problem. This model is then used to compute quickly the associated failure rate. [1] [2]. In the second category, hybrid methodologies rely both on a model of the problem and on MC simulations as well. For example, the authors in [3] and [4] use a classifier to screen out the circuit run points that do not significantly contribute to the failure rate estimation, before making the time-consuming simulations. The last category is based on Importance Sampling (IS), which intentionally modifies the distribution of the random transistor parameters to generate only circuit run points that significantly contribute to the failure rate estimation. The challenge in IS is to find the proper distribution of the random transistor parameters. In this paper, we propose the Gradient Importance Sampling (GIS) methodology using gradient-descent to find these distributions.

The remainder of this paper is organized as follows. Section II gives a summary of MC and IS methodology. State-of-the-art techniques for high-sigma statistical extraction of circuit characteristics are then briefly described. Section III explains the proposed GIS methodology. Section IV provides details on its application to the assessment of the write latency and the read access time of a column in a static random access memory (SRAM). In Section V, we study the impact of the methodology parameters to optimize the convergence speed, which is compared to the SoA in Section VI.

II. HIGH-SIGMA EXTRACTION METHODOLOGIES FOR CIRCUIT DESIGN

The proposed GIS algorithm relies on Importance Sampling. Therefore, a brief overview of standard MC and IS will be given first.

A. Mathematical formulation of Monte Carlo methodology

The MC methodology is one of the most popular statistical simulation methodology. It is suitable for solving high-dimensional problems compared to analytical methodologies
and can be used to estimate a probability of a certain event \( Y \). The principle of MC is as follows: Usually the event of interest, e.g. the failure of a given circuit, depends on a M-dimensional random variable \( X \) distributed as \( f(X) \), which in the circuit case are the individual transistor parameters such as their \( V_t \), gate oxide length, mobility, etc. The goal is to find the probability \( p \) such that:

\[
p = P_X \{ \theta(X) \in Y \} \tag{1}
\]

where \( \theta(X) \) is a function corresponding to a circuit characteristic, which depends on the transistor parameters. \( Y \) is a subset of \( \theta \) values corresponding to a failure. Both \( X \) and \( \theta \) can be multivariate quantities, and the dimension of neither need to be fixed. This probability \( p \) can also be seen as an integral:

\[
p = \mathbb{P}_X [\theta(X) \in Y] = \mathbb{E}_F [1(\theta(X) \in Y)] = \int_{\mathbb{R}^M} 1(\theta(X) \in Y) f(X) \, dX \tag{2}
\]

where \( 1(Z) \) is the truth-indicator and takes the value 1 if \( Z \) is true and 0 otherwise. MC simulations generate \( N \) independent samples of \( X \) from \( f(x) \) called \( x_1, \ldots, x_N \), producing \( \theta_i = \theta(x_i) \). The observed values \( \theta_i \) are used as an estimate of the distribution of \( \theta(X) \), with equal weights. An estimator of the probability \( p \) denoted by \( \hat{p}_{MC} \) can be computed as the fraction of evaluated samples belonging to event \( Y \):

\[
\hat{p}_{MC} = \frac{1}{N} \sum_{i=1}^{N} 1(\theta(x_i) \in Y) \tag{3}
\]

The estimator of the variance of the MC estimator is given by [6]:

\[
\text{VAR}(\hat{p}_{MC}) = \frac{\hat{p}_{MC}}{N} (1 - \hat{p}_{MC}) \approx \frac{\text{var}(p)}{N} \text{ for small } \hat{p}_{MC} \tag{4}
\]

In [6] [11], the figure of merit (FoM) for the convergence rate is defined as:

\[
\rho(\hat{p}) = \frac{\sqrt{\text{var}(p)}}{\hat{p}} \tag{5}
\]

In fact, the FoM \( \rho \) can be treated as a relative error so that a smaller FoM means higher accuracy. One quantity of interest is the number of samples required to reach an estimator of \( p \) with a given confidence of \( (1 - \delta) \times 100% \) and a given accuracy \( (1 - \varepsilon) \times 100% \), i.e.

\[
P \left( \frac{\hat{p}_{MC} - p}{p} \leq \delta \right) = 1 - \varepsilon \tag{6}
\]

For large \( N \) and by using the central limit theorem (CLT), the number of samples required is given by:

\[
N(\varepsilon, \delta) \approx \frac{Q(1 - \frac{\varepsilon}{2})}{\delta^2} \tag{7}
\]

where \( Q \) is the quantile function. Thus, for an accuracy and a confidence of 90\%, roughly \( \frac{220}{\varepsilon^2} \) samples are required. For rare event (i.e. low \( p \)), a very long sequence is required and thus the simulation necessitates a large computation time. Attempting to obtain a low variance estimate of \( p \) by increasing \( N \) is therefore not efficient.

\section*{B. Mathematical formulation of Importance Sampling methodology}

MC is of limited use to detect rare events because it takes on average \( 1/p \) samples to observe a failure. To overcome the MC speed limitation, one could change the distribution \( f(x) \) of \( X \) such that the event \( Y \) becomes more likely under an alternative distribution \( g(x) \), usually referred to as a biasing distribution. A reliable estimate of \( p \) can be obtained with fewer samples as many of them fall in the region of interest. The original probability \( p \) is found by weighting the alternative probability function of \( f(x) \) and \( g(x) \). This is the key idea behind IS. More formally, Eq. (2) becomes:

\[
p = \int_{\mathbb{R}^M} 1(\theta(X) \in Y) f(X) \, dX = \int_{\mathbb{R}^M} 1(\theta(X) \in Y) \frac{f(X)}{g(X)} g(X) \, dX = \mathbb{E}_{g_X} [w_X(1(\theta(X) \in Y))] \tag{8}
\]

where \( w(x) = f(x)/g(x) \) is a likelihood ratio and is referred to as a weight function. IS simulations generate \( N \) independent samples of \( X \) from \( g(x) \) denoted \( x_1, \ldots, x_N \). The new estimator of the probability \( p \) denoted as \( \hat{p}_{IS} \) is obtained as follows:

\[
\hat{p}_{IS} = \frac{1}{N} \sum_{i=1}^{N} w(x_i) 1(\theta(x_i) \in Y) \tag{9}
\]

The estimator of the variance of IS is given by [6]:

\[
\text{VAR}(\hat{p}_{IS}) = \frac{1}{N^2} \left( \sum_{i=1}^{N} w(x_i)^2 1(\theta(x_i) \in Y) - N\hat{p}_{IS}^2 \right) \tag{10}
\]

However, IS makes sense only if it converges at a faster rate than MC (note that MC is special case of IS where \( g(x) = f(x) \) i.e. \( w(x) = 1 \)). Given our requirement of an accuracy of \( (1 - \delta) \times 100\% \) and a confidence of \( (1 - \varepsilon) \times 100\% \), we must ensure that \( \rho(\hat{p}_{IS}) \) converges at a faster rate than \( \rho(\hat{p}_{MC}) \). The question is which biasing distribution \( g(x) \) is to be used to ensure fast convergence?

\section*{C. State-of-the-art}

As mentioned in the introduction, in IC design, the transistors suffer from random variations during fabrication. They are usually modelled as a shift of the \( V_t \), gate oxide thickness, mobility, critical dimensions (width and length), etc. In the transistor compact models, these shifts are distributed accordingly to a normal distribution. During MC simulations, samples are generated for each transistor and depending on the selected characteristic of the circuit (i.e. the selected \( \theta(X) \)), the failure rate is computed accordingly to Eq. (3).

Before explaining the proposed algorithm, let us have a look at the state of the art in low failure rate extraction. Many methodologies rely on the concept of IS. The efficiency of previously-proposed IS methodologies is heavily dependent on
the biasing distribution used. Methodologies like Mixture ratioed Importance Sampling [5] replace the normal distribution of the $V_i$ by a combination of uniform and normal distributions. Other methodologies simplify the approach by altering the initial normal distribution. The mean of these distributions can be shifted to the failing point closest in a quadratic distance to the nominal operating point. Such point is called the most probable failure point (MPFP) and corresponds to the most likely failure mechanism [7]. Importance Sampling augmented with MPFP search has been explored in [7], [8] and [9]. Our algorithm is also based on this approach. However, many of the existing methodologies are not very efficient at finding the MPFP [9]. In [7], [8] and [9], the algorithm requires at each iteration to sample an hyper-sphere and simulate it. This is inefficient because at each iteration, amongst all the points randomly sampled, only one point is kept [7] [8] or a few points are kept [9] [11] and every other points are discarded without contributing to the knowledge of the failure region for the next iteration. In addition, these methodologies do not scale well for high-dimensional problems: given a maximum distance between two consecutive samples, the number of point required to sample an N-dimensional hypersphere depends exponentially on N and on N-1 when sampling the surface of this hypersphere. All these methodologies spend more or less time trying to find their optimal biasing distribution $g(x)$. In other words, there is a trade-off between the number of simulation runs spent to find $g(x)$ i.e. searching for the MPFP and the number of runs actually spent to compute the probability $p$.

### III. Proposed methodology: Gradient Importance Sampling

The gradient importance sampling algorithm aims to efficiently find the MPFP with a minimum number of simulation runs. It relies on a gradient descent of the simulated circuit characteristic of interest $\theta$. More formally, let us consider a circuit with an M-dimension $X$ vector corresponding to the $V_i$ of its $M$ transistors where one wants to assess the probability to reach a given circuit characteristic $\theta_{obj}$. The proposed algorithm works as follows.

**Step 1. Find the MPFP via a gradient-based search**

- Initialize the index of iteration $n_{iter} = 1$, the M-dimensional vector $s = \{0\}^M$ which represents the shift of the mean of the $V_i$ distributions, the norm of the gradient-descent step on $s$ step = 1 and the minimum step size step$_{min}$ = 1%.
- Let $\mu$ and $\sigma$ be the vector of the mean and the standard deviation of the $V_i$ normal distributions.
- Compute the nominal value of the circuit characteristic $\theta_0 = \theta(\mu)$.
- while (step > step$_{min}$)
  1) Extract the gradient of $\theta_{iter-1}$ by computing independently the derivative with respect to the $V_i$ of each transistor with its $\Delta V_i$ normalized by the corresponding $\sigma$ (see Fig. 1). In term of complexity, one simulation is needed for each dimension of the problem plus one for computing $\theta(n_{iter} - 1)$.
  2) Find the vector $\nu$ of norm step with the steepest positive slope based on the gradient $\frac{\Delta \theta_{iter}}{\Delta V_i}$.
  3) Simulate the circuit characteristic $\theta_{iter} = \theta(\mu + s + \nu)$.
  4) if ($\theta_{iter} > \theta_{obj}$)
      set $s = s + \nu$
      else if ($\theta_{iter} < \theta_{obj}$)
      set step = step/2 and go back to 2)
  5) $n_{iter} = n_{iter} + 1$.

Depending on the problem or the objective, one might need to compute the steepest negative slope. In this case, the signs of the last if else if loop need also to be reverted.

Fig. 1 shows an example of what the gradient exploration might look like in a two dimensional space. The resulting vector $s$ is computed at each iteration. When crossing the failure region, the distance step is divided by two. The loop is iterated until step reaches 0.01$\sigma$. A study of the impact of this parameter is performed in Section V based on the testbench described in Section IV. The iterations will eventually lead us to the MPFP.

**Step 2. Run importance sampling simulations** Once the MPFP is found, run an importance sampling simulations with the mean of the normal distributions of the $V_i$ shifted by the vector $s$. To compute the failure rate, the weights $w(x)$ are given in the case of mean shift IS by:

$$w(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$= \exp\left(-s\left(2\mu - s - 2\mu\right)\right)$$

and must be applied in Eq. (9). After a variable number of IS runs, the estimator reaches the required value of $\rho$. The
total number of runs required to find the failure rate at $\theta_{obj}$ is is the sum of MPFP runs and the IS runs.

The proposed MPFP search algorithm presents a few pros and cons compared to the SoA. As mentioned earlier, each iteration only needs one simulation per dimension plus one nominal simulation. The complexity of the gradient exploration is thus in $\mathcal{O}(M)$. On the other hand the main drawback of the gradient algorithm is that it only works with convex problems. Otherwise it can get stuck in a local extremum. It is possible to avoid this lock by fine tuning the initial parameter step but it must be done on a case by case basis. This parameter will also strongly influence the precision and the number of iterations required to find the MPFP. A study of the impact of this parameter step is carried out in Section V.

IV. APPLICATION TO HIGH-SIGMA SRAM DYNAMIC CHARACTERISTIC EXTRACTION

In this section, we will see how this algorithm can be used to extract low failure rate in SRAM design. Indeed, as SRAM arrays are designed with minimum feature size they are strongly affected by the inevitable presence of random variations. The proposed algorithm is used to estimate the yield with respect to dynamic characteristics of a column of 256 standard 6T SRAM bitcells with a sense amplifier (SA) as shown in Fig. 2.

Two use-cases are defined to monitor the effect of the dimensionality on the algorithm. The first use case assesses the write latency time (WLT) of the bitcell when a logic 1 is written on Q. This problem is in 6 dimensions as we consider ideal drivers (i.e. only the 6 transistors of the bitcell are monitored). The second use case concerns the read access time (RAT) of the column when a logic 0 is read from Q. For this problem, the SA and the write transistor of the 255 unaccessed bitcells are also taken into account. The worst case is considered: the other 255 bitcells store a logic 1. A single $V_t$ shift can be used for these unaccessed bitcells as the transistors can be considered as a single transistor $255 \times$ wider. The second problem is thus a 12 dimensional problem.

All simulations are performed in 28nm FDSOI technology with High-k Metal Gate at 0.45V at the nominal process corner and at room temperature. Unless mentioned otherwise, a confidence and an accuracy of 90% are considered for the computation of the convergence FoM (i.e $\rho = 0.1$).

The results of the proposed algorithm are shown in Table I for both use cases. From a circuit point of view the results are in accordance with the intuition a designer might have.

- In the case of the WLT, a logic 1 is written on Q. As the access NMOS $Acc_0$ does not pull up QB well, the PMOS $P_{d0}$ is under stress because it needs to fully restore the voltage on QB. This explains the high positive $V_t$ shift applied to $P_{d0}$ and the smaller positive $V_t$ shift on $Acc_0$.
- In the case of the RAT, a logic 0 is read from Q. This time, it is the opposite. The access transistor $Acc_0$ must discharge as fast as possible the BL and the $P_{d0}$ must keep Q at 0V. Nothing much appends on QB as the BLs are precharged to $V_{DD}$ before a read. The $V_t$ of $Acc_{ub}$ transistors are a bit decreased so that they will leak and discharge QB while the opposite is true for the $V_t$ of $Acc_u$. Finally, as the SA is sized with wide transistors, no significant shifts are applied to the $V_t$.

Thus, the transistors with the highest shift of $V_t$ are the most critical transistors in the design in terms of variability. If a designer does not meet the design target, he knows which part of the circuit needs to be resized. The evolution of the estimated failure probability is shown in Fig. 3 and Fig. 4 versus the simulation run index for both the importance sampling part of the proposed GIS and the standard MC. For both the WLT and the RAT, we can see that GIS and MC converge toward the same failure rate. Table II shows the total number of runs required for different failure rates for the two use cases. The first thing to notice is that as expected the number of runs required for the gradient exploration does not depend too much on the failure rate but mainly on the dimensionality of the problem. Regarding the speed-up, the lower the failure rate, the greater the gain.
TABLE I
SHIFT VECTOR APPLIED TO THE BITCELL FOR A WLT OF 4NS IN THE CASE OF THE WLT AND 10 NS IN THE CASE OF THE RAT.

<table>
<thead>
<tr>
<th>Transistor</th>
<th>$P_{d0}$</th>
<th>$P_{d1}$</th>
<th>$P_{d2}$</th>
<th>$P_{d3}$</th>
<th>$Acc_{0}$</th>
<th>$Acc_{1}$</th>
<th>$Acc_{2}$</th>
<th>$Acc_{3}$</th>
<th>$PS_{A1}$</th>
<th>$N_{A1}$</th>
<th>$PS_{A2}$</th>
<th>$N_{A2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>WLT VT shift [mV]</td>
<td>102.37</td>
<td>-0.81</td>
<td>-2.35</td>
<td>-0.65</td>
<td>10.48</td>
<td>9.48</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RAT VT shift [mV]</td>
<td>-2.21</td>
<td>44.0</td>
<td>-0.69</td>
<td>-0.79</td>
<td>24.7</td>
<td>-0.29</td>
<td>1.95</td>
<td>-1.68</td>
<td>-0.11</td>
<td>-0.17</td>
<td>0.68</td>
<td>-2.28</td>
</tr>
</tbody>
</table>

V. IMPLEMENTATION CHOICES

Let us now discuss the algorithm implementation choices that were made.

1) When crossing the failure region, two options arise. First, we can perform either a classical dichotomic search as in [7]. After crossing the border, the direction of the slope can be reversed. Then a new gradient is computed with $step = step/2$ and the vector $s$ is updated. The second option is to only recompute the vector $s$ with $step = step/2$. The second option is used in our implementation and offers the advantage of only requiring one run where as the first one requires $N+1$ runs.

2) The minimum step size $min_{step}$ is set arbitrarily at $1\% \sigma$ in [9] and [12]. This value was also used in our baseline implementation. Fig. 5 shows the evolution of the number of runs required for the gradient exploration and IS to reach a FoM of 0.1 (as described in Section IV) in function of $min_{step}$. Two trends can be observed. Firstly as $min_{step}$ decreases, the number of runs required to find the MPFP increases more or less exponentially. Secondly, the number of IS runs keeps decreasing until it reaches a plateau. In the case of the WLT assessment, an optimum is found around $10^{-5}$ for a failure rate of $\approx 1 \times 10^{-9}$. This optimum value can vary from one use-case to another but Fig. 5 indicates that the sensitivity of the algorithm to $min_{step}$ is limited.

3) The last important parameter is the initial value of the step $step$ used for the first iterations. As explained earlier if it is chosen too small the algorithm will need many iterations to converge but the results will be more accurate. On the other hand if chosen too large, the algorithm will converge quickly but the result might not be optimal. Fig. 6 shows the evolution of the number of runs required for the algorithm to converge decreases as $step$ increases. Secondly, more and more runs are needed when $step$ increases as the point found by the algorithm is less accurate. In the case of the WLT assessment, an optimum is found around $1 \sigma$ for a failure rate of $\approx 10^{-9}$. Again this is use-case dependent but the sensitivity is limited.

VI. CONVERGENCE SPEED: GIS VERSUS THE SOA

Let us make a comparison of GIS to the SoA techniques. Fig. 7 shows the number of runs required to extract different
failures rate with the SoA techniques. We can observe that GIS is the best-in-class algorithm for low failure rate extraction. This is due to the fact the MPFP search in the proposed methodology is very efficient and straightforward compared to other techniques. Some methodologies such as [10] yield a more accurate biasing distribution but the number of runs to compute this distribution is higher than the entire number of runs required for GIS and MC.

VII. Conclusion

In this paper, we presented “Gradient Importance Sampling”: a efficient methodology based on mean shift Importance Sampling that can determine the MPFP. Rather than sampling random points, the method relies on the gradient of the observed metric to quickly find the MPFP. We showed for the extraction of write latency and read access time in a 256-bitcell SRAM column that GIS yields the best results for low failure rate extraction.

Acknowledgements

This work has been funded by the European Union under EU Framework Programme 7, FP7-ICT-2013-10, MSP-Project Grant Agreement No. 611887 and by the SAVE project within the frame of the Plan Marshall of the Walloon region of Belgium.

References