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

Thomas Haine[1], Johan Segers[2], Denis Flandre[1], David Bol[1]
[1]*ICTEAM Institute,* [2]*Institute 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 3.8×10^{-9}. Therefore, on average 3.8×10^{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

978-3-9819263-0-9/DATE18/©2018 EDAA

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 length, mobility, etc. The goal is to find the probability p such that:

$$p = \mathbb{P}_{f_X}\left[\theta(X) \in Y\right] \tag{1}$$

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

$$\begin{aligned} p &= \mathbb{P}_{f_X}\left[\theta\left(X\right) \in Y\right] \\ &= \mathbb{E}_{f_X}\left[1\left(\theta\left(X\right) \in Y\right)\right] \\ &= \int_{\mathbb{R}^M} 1\left(\theta\left(X\right) \in Y\right) f\left(X\right) dX \end{aligned} \tag{2}$$

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, ..., x_n producing $\theta_i = \theta(x_i)$. The observed values θ_i are used as an estimate of the distribution of $\theta(X)$, with equal weights. An estimator of the probability p denoted by $\widehat{p}_{MC}$ can be computed as the fraction of evaluated samples belonging to event Y:

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

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

$$\begin{aligned} \widehat{VAR}\left(\widehat{p}_{MC}\right) &= \frac{\widehat{p}_{MC}}{N}\left(1 - \widehat{p}_{MC}\right) \\ &\approx \frac{\widehat{p}_{MC}}{N} \text{ for small } \widehat{p}_{MC} \end{aligned} \tag{4}$$

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

$$\rho(\widehat{p}) = \frac{\sqrt{var\left(\widehat{p}\right)}}{\widehat{p}} \tag{5}$$

In fact, the FoM ρ 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.

$$P\left(\frac{|\widehat{p}_{MC} - p|}{p} \leq \delta\right) = 1 - \varepsilon \tag{6}$$

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

$$N(\varepsilon, \delta) \approx \frac{Q\left(1 - \frac{\varepsilon}{2}\right)}{\delta p^2} \tag{7}$$

where Q is the quantile function. Thus, for an accuracy and a confidence of 90%, roughly $\frac{270}{p}$ 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.

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 in function of $f(x)$ and $g(x)$. This is the key idea behind IS. More formally, Eq. (2) becomes :

$$\begin{aligned} p &= \int_{\mathbb{R}^M} 1\left(\theta\left(X\right) \in Y\right) f\left(X\right) dX \\ &= \int_{\mathbb{R}^M} 1\left(\theta\left(X\right) \in Y\right) \frac{f\left(X\right)}{g\left(X\right)} g\left(X\right) dX \\ &= \mathbb{E}_{g_X}\left[w_X\left(X\right) 1\left(\theta\left(X\right) \in Y\right)\right] \end{aligned} \tag{8}$$

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, ..., x_n. The new estimator of the probability p denoted as $\widehat{p}_{IS}$ is obtained as follows:

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

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

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

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 $\rho(\widehat{p}_{IS})$ converges at a faster rate than $\rho(\widehat{p}_{MC})$. The question is which biasing distribution $g(x)$ is to be used to ensure fast convergence?

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 transistors 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_t 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 θ. More formally, let us consider a circuit with an M-dimension X vector corresponding to the V_t of its M transistors where one wants to assess the probability to reach a given circuit characteristic θ_{obj}. The proposed algorithm works as follows.

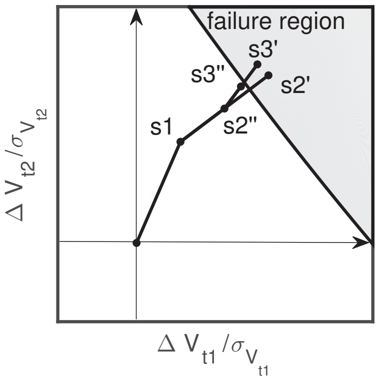
failure region
s3'
s3''
s2'
s1
s2''
$\Delta V_{t2}/\sigma_{V_{t2}}$
$\Delta V_{t1}/\sigma_{V_{t1}}$

Fig. 1. Illustration of MPFP search algorithm in the proposed GIS methodology in a two dimensional space. When crossing the failure region, the distance d is divided by 2 and the metric is recomputed until this new metric stays in the pass region.

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_t distributions, the norm of the gradient-descent step on s $step = 1$ and the minimum step size $step_{min} = 1\%$.
- Let μ and σ be the vector of the mean and the standard deviation of the V_t normal distributions.
- Compute the nominal value of the circuit characteristic $\theta_0 = \theta(\mu)$.
- **while** $(step > step_{min})$
  1) Extract the gradient of $\theta_{n_{iter}-1}$ by computing independently the derivative with respect to the V_t of each transistor with its ΔV_t normalized by the corresponding σ (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 v of norm $step$ with the steepest positive slope based on the gradient $\frac{\Delta\theta_{n_{iter}-1}}{\Delta V_t}$.
  3) Simulate the circuit characteristic
     $\theta_{n_{iter}} = \theta(\mu + s + v)$.
  4) **if** $(\theta_{n_{iter}} > \theta_{obj})$
        set $s = s + v$
     else if $(\theta_{n_{iter}} < \theta_{obj})$
        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 0.01σ. 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_t shifted by the vector s. To compute the failure rate, the weights $w(x)$ are given in the case of mean shift IS by :

$$\begin{aligned} w(x) &= \frac{\frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)}{\frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu-s)^2}{2\sigma^2}\right)} \\ &= \exp\left(\frac{-s\,(2x - s - 2\mu)}{2\sigma^2}\right) \end{aligned} \tag{11}$$

and must be applied in Eq. (9). After a variable number of IS runs, the estimator reaches the required value of ρ. The total number of runs required to find the failure rate at θ_{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 × 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_{0b} does not pull up QB well, the PMOS P_{u0} is under stress because it needs to fully restore the voltage on QB. This explains the high positive V_t shift applied to P_{u0} and the smaller positive V_t shift on Acc_{0b}.
- 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.

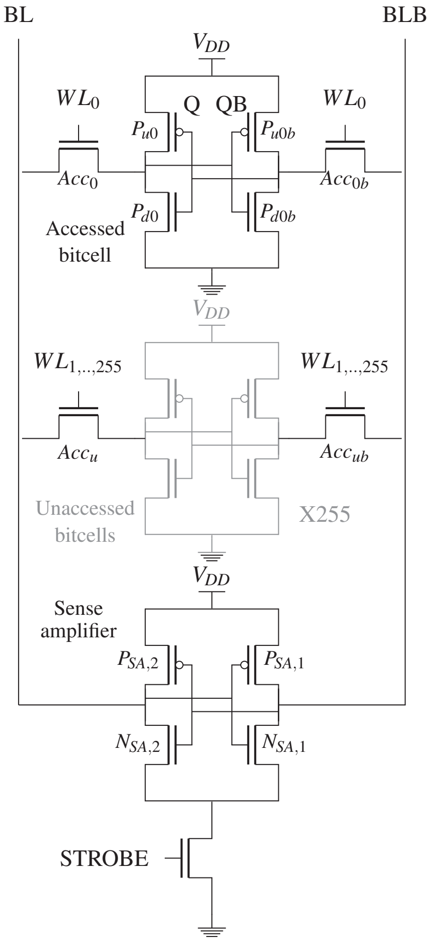

Fig. 2. Transistor level schematics of a representative memory column with 256 bitcells per bitline.

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.

Transistor	P_{u0}	P_{d0}	P_{u0b}	P_{d0b}	Acc_0	Acc_{0b}	Acc_u	Acc_{ub}	$P_{SA,1}$	$N_{SA,1}$	$P_{SA,2}$	$N_{SA,2}$
WLT VT shift [mV]	102.37	-0.81	-2.35	-0.65	10.48	9.48	-	-	-	-	-	-
RAT VT shift [mV]	-2.21	44.0	-0.69	-0.79	24.7	-0.29	1.95	-1.68	-0.11	-0.17	0.68	-2.28

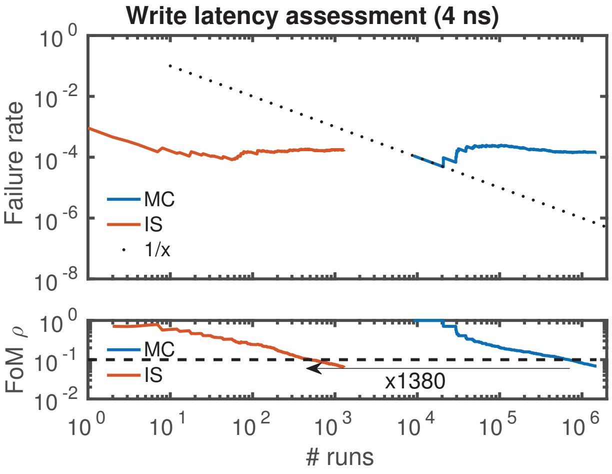

Fig. 3. (a) Evolution of failure probability estimate from GIS compared with baseline MC for a 4ns write latency of the bitcell as depicted in Fig. 2. (b) Evolution of the FoM of convergence in function of the number of runs. A speed-up of 1.38×10^3 is achieved compared to MC when considering $\rho = 0.1$.

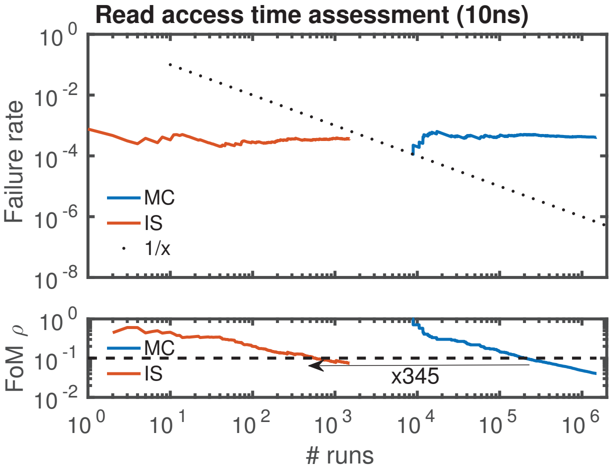

Fig. 4. (a) Evolution of failure probability estimate from GIS compared with baseline MC for a 10ns read access time of a bitcell as depicted in Fig. 2. (b) Evolution of the FoM of convergence in function of the number of runs. A speed-up of 345 is achieved compared to MC when considering $\rho = 0.1$.

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.

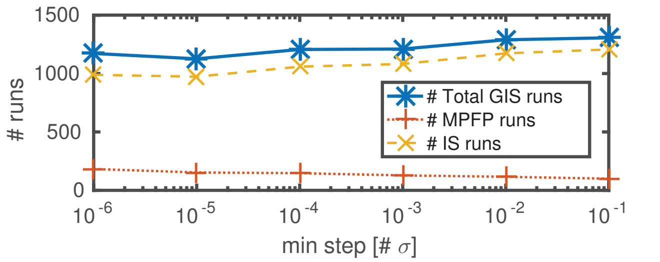

Fig. 5. Evolution of the number of runs in function of the minimum step (in term of σ). The testbench used was the WLT assessment with a failure rate of $\approx 1 \times 10^{-9}$.

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 $\approx 10^{-9}$. 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 gradient exploration and IS to reach an FoM of 0.1 (as described in Section IV) in function of $step$. Again two trends can be observed. First as expected 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 1σ 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

TABLE II

SUMMARY OF GIS SIMULATIONS RESULTS ON SPICE SIMULATIONS OF THE (A) WRITE LATENCY OF A 6T BITCELL (6 DIMENSIONS) AND (B) THE READ ACCESS TIME AS DEPICTED IN FIG. 2 (12 DIMENSIONS). FOR ALL SIMULATIONS A CONVERGENCE OF $\rho = 0.1$ IS CONSIDERED.

	WLT	WLT	WLT	RAT	RAT	RAT
Dimensions	6	6	6	12	12	12
Failure rate	1.7×10^{-4}	4.6×10^{-7}	2.6×10^{-9}	4×10^{-4}	1×10^{-5}	1.4×10^{-7}
# MPFP runs	80	84	116	240	283	332
# IS runs	496	842	1075	544	1070	1556
# Total GIS runs	576	927	1191	784	1355	1888
Speed-up	1.19×10^3	6.3×10^5†	8.7×10^7†	253	1.98×10^4†	1×10^6†

† Projections on the number of MC runs using the rule of thumb of 270/failure rate

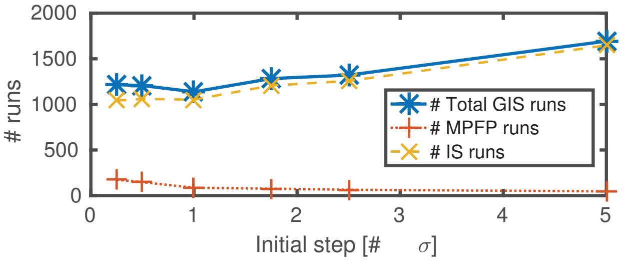

Fig. 6. Evolution of the number of runs in function of the length of the initial step *step* in term of sigma. The testbench used was the WLT assessment with a failure rate of ≈ 1e-9.

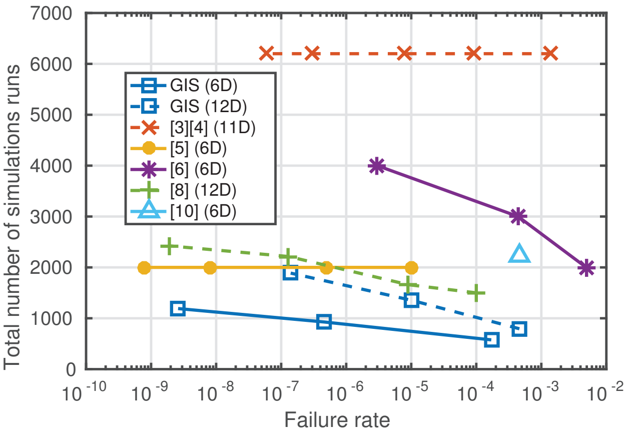

Fig. 7. Simulation cost comparison to other methodologies for low failure rate evaluation. The proposed algorithm converges faster than any other methodology. Updated version of the figure in [8].

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

[1] J. Boley, V. Chandra, R. Aitken and B. Calhoun,*"Leveraging Sensitivity Analysis for Fast, Accurate Estimation of SRAM Dynamic Write VMIN"*, in IEEE DATE, 2013.

[2] P. Weckx, B. Kaczer, H. Kukner, Ph. J. Roussel, P. Raghavan, F. Catthoor, G. Groeseneken, *"Non-Monte-Carlo methodology for high-sigma simulations of circuits under workload-dependent BTI degradation-application to 6T SRAM"*, in IEEE ISRP, 2014.

[3] A. Singhee and R. Rutenbar,*"Statistical blockade: A novel method for very fast monte carlo simulation of rare circuit events, and its application"*, in Proc. Design Autom. Test Eur. Conf. Exhib., pp. 1-6, 2007.

[4] A. Singhee and R. Rutenbar,*"Statistical blockade: Very fast statistical simulation and modeling of rare circuit events and its application to memory design"*, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 28, no. 8, pp. 1176-1189, 2009.

[5] R. Kanj, R. Joshi, and S. Nassif,*"Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events"*, in Proc. IEEE/ACM Design Autom. Conf., pp. 69-72, 2006.

[6] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan,*"Breaking the simulation barrier: SRAM evaluation through norm minimization"*, in Proc. Int. Conf. IEEE Comput.-Aided Design, pp. 322-329, 2008.

[7] M. Qazi, M. Tikekar, L. Dolecek, D. Shah and A. Chandrakasan,*"Loop Flattening & Spherical Sampling: Highly efficient Model Reduction Techniques for SRAM yield Analysis"*, in IEEE DATE, 2010.

[8] M. Qazi, M. Tikekar, L. Dolecek, D. Shah and A. Chandrakasan,*"Technique for Efficient Evaluation of SRAM Timing Failure"*, in IEEE TVLSI, vol 21, no 8, pp 1558-1562, 2013.

[9] M. Rana and R. Canal, *"SSFB: A Highly-Efficient and Scalable Simulation Reduction Technique for SRAM Yield Analysis"*, in IEEE DATE, 2014.

[10] F. Gong, S. Basir-Kazeruni, L. Dolecek and L. He,*"A Fast Estimation of SRAM Failure Rate Using Probability Collectives"*, in Proceedings of the International Symposium on Physical Design, pp. 41-48, 2012.

[11] K. Katayamay, S. Hagiwarayy, H. Tsutsuiy, H. Ochiy and T. Satoy,*"Sequential Importance Sampling for Low-Probability and High-Dimensional SRAM Yield Analysis"*, in IEEE/ACM ICCAD ,2010.

[12] J. Jaffari and M. Anis,*"Adaptive Sampling for Efficient Failure Probability Analysis of SRAM Cells"*, in IEEE/ACM ICCAD ,2009.