An Efficient SRAM yield Analysis Using Scaled-Sigma Adaptive Importance Sampling

[1,2]Liang Pang, [1]Mengyun Yao, [1]Yifan Chai
[1]School of Microelectronics, Southeast University, Nanjing, 210096;
[2]School of Electronic Science & Engineering, Southeast University, Nanjing, 210096
* Email: 230189800@seu.edu.cn , yaomengyun89@163.com , 220184780@seu.edu.cn

***Abstract*—Statistical SRAM yield analysis has become a growing concern for the requirement of high integration density and reliability of SRAM under process variations. It is a challenge to estimate the SRAM failure probability efficiently and accurately because the circuit failure is a "rare-event". Existing methods are still not efficient enough to solve the problem, especially in high dimensions. In this paper, we develop a scaled-sigma adaptive importance sampling (SSAIS) which is an extension of the adaptive importance sampling. This method changes not only the location parameters but the shape parameters by searching the failure region iteratively. The 40nm SRAM cell experiment validated that our method outperforms Monte Carlo method by 1500x and is 2.3x~5.2x faster than the state-of-art methods with reasonable accuracy. Another experiment on sense amplifier shows our method achieves 1811x speedup over the Monte Carlo method and 2x~11x speedup over the other methods.**

I. INTRODUCTION

As semiconductor technology continues to advance, process fluctuations have an increasing impact on circuit reliability, especially for SRAM. The design of SRAM is always moving towards high density and high performance so that the size of memory cells is usually designed to be as small as possible [1]. As a result, smaller sizes make SRAM cells more susceptible to process fluctuations [11]. Hence, it is critical to concern about the failure rate of SRAM to obtain a robust SRAM. However, SRAM cells are typically replicated millions of times in a memory system. The failure rate of each SRAM cell should be extremely low (10^{-8}~10^{-6}) to make sure the acceptable yield of the whole SRAM. For such a "rare-event" estimation, the analytical method is infeasible.

To estimate the failure rate of SRAM accurately, many statistical methods have been proposed. Among them, Monte Carlo (MC) method remains the gold standard which samples the variation space directly and simulates each sample to get the corresponding circuit performance. However, MC approach is extremely time-consuming in estimating SRAM failure rate because it needs hundred millions of simulations to get a convergent result. To accelerate MC method, related work can be grouped into two categories:

Classification: Statistical Blockade (SB) [3] applies a "classifier" to classify more likely-to-fail samples and only run these samples. The recently work [4] has improved the efficiency of classification by constructing a conditional classifier and SVM-based nonlinear classifier respectively. However, the main drawbacks are still unsolved that training an accurate classifier needs too many samples when the required failure rate is extremely low.

Importance Sampling: The basic idea of importance sampling methods is to find a distort sampling distribution to sample near the failure region. The efficiency and accuracy of IS methods heavily depend on distort probability distribution. Different approaches have been proposed to construct such a distribution. MixIS [6] mixes the original distribution, the uniform distribution and the shifted distribution centered around the collected failure samples. HDIS [7] shift the mean to center of failure region. MNIS [8] and SS [9] shift the mean to the boundary of failure region by a minimum L2-norm. In order to evaluate the impact of multiple indicators comprehensively, there are several methods, such as [10][12], to construct a mixture of multiple mean-shift distributions by shifting the mean vectors to various failure regions caused by different performance metrics. However, the methods above become inefficient in high dimension scenario when estimating dynamic metrics of SRAM cell. Because it is very difficult to determine the failure boundary in high-dimension variation space.

Sequential Importance Sampling (SIS) [13] uses a resampling technique to decrease the samples consumed for searching failure boundary. And Adaptive Importance Sampling (AIS) [2] updates an unbiased estimator along with resampling iterations to further decrease the samples consumed in the static IS step. However, both of them use a heuristic predefined variance of the prediction distribution so that their efficiency and stability are suffered from the choice of the optimal variance.

In this paper, we developed a method based Scaled Sigma Adaptive Importance Sampling (SSAIS) to estimate the failure rate of SRAM stably and accurately, which is an extension of AIS [2]. We first collect a few failure samples to construct an initial distort sampling distribution with its mean and variance are defined as location parameters and shape parameters. Next, the proposed methodology will iteratively move the mean of current sampling distribution to the high probability failure region by resampling technique. Then, after a batch of iterations, we calculate the shape parameters mathematically to start iteration in the next batch. The failure rate of SRAM will be updated by unbiased estimator iteratively to eliminate the samples of the IS step of conventional IS based methods.

The AIS [2] shows the huge difference in efficiency and accuracy with different variations. Although the algorithm with small variance can reach a reasonable result as soon as possible, it completely failed in estimating the failure rate finally due to the diversity of samples decreases greatly as the iteration continues. We extend AIS in two significant way: 1) we proposed a solution to update an optimal variance after completing a batch which can accelerate the convergence speed with exploring the failure region fully 2) we divide the whole iteration process into several batches, and each batch contain only a small number of iterations to improve the efficiency without wasting additional simulations as much as possible.

The rest of this paper is organized as follows. In Section 2, the rare event analysis problem and static importance sampling approach are revisited. Section 3 provides details about our algorithm. The accuracy and efficiency will be demonstrated by Several experiments in Section 4. In Section 5, we will give our conclusion finally.

II. Background

A. Problem Formulation

We consider $X = \{x_1, x_2, \dots, x_m\}$ as a vector consisted of m independent Gaussian variables modeled from process parameters. And $f(X)$ is the probability density function (PDF) of X. Let Y be certain performance metric which can be measured through expensive transistor-level simulation, such as SRAM read voltage, write time, etc.

For the failure rate evaluation of SRAM, we denote S is the failure region. Typically, the set S is extremely tiny. We define the circuit performance doesn't meet the specification when $Y \in S$. And we further introduce indicator function $I(X)$ to identify pass/fail of Y:

$$I(X) = \begin{cases} 0, & if \quad Y \notin S \\ 1, & if \quad Y \in S \end{cases} \tag{1}$$

Therefore, the probability can be calculated as:

$$P_{fail} = P(Y \in S) = \int I(X) \cdot f(X) dX \tag{2}$$

Unfortunately, the formulation (2) is difficult to calculate analytically because we don't know what distribution $I(X)$ satisfies exactly. Traditionally, Monte Carlo is used to estimate the failure probability by sampling from $f(X)$ directly, and the unbiased estimate of P_{fail}:

$$\hat{P}_{fail} = \hat{P}(Y \in S) = \frac{1}{N}\sum_{i=1}^{N} I(x_i) \xrightarrow{N \to +\infty} P(Y \in S) \tag{3}$$

B. Importance Sampling

For estimating SRAM cell failure rate, $Y \in S$ is a rare event. Standard MC requires hundreds of million samples to ensure that the results are gradually converged. It is not realistic to apply MC to the practical design especially the circuit size is large because each sample needs a transistor-level simulation. To address this issue, importance sampling has been introduced to sample near the failure region through a "distorted" sampling distribution $g(X)$.

As shown in the one-dimensional case shown in Fig 1, by constructing the distorted sampling function $g(x)$, we can pick more failure samples. And the failure probability can be expressed as (4):

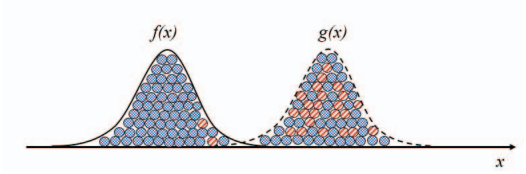

Fig 1. Mean-shift Importance Sampling

$$P_{fail} = P(Y \in S) = \int I(X) \cdot \frac{f(X)}{g(X)} \cdot g(X) dX \tag{4}$$

$$= \int I(X) \cdot w(X) \cdot g(X) dX \tag{5}$$

Here, the $w(x)$ denotes the likelihood ratio between original PDF $f(x)$ and the distort PDF $g(x)$ which compensates for the discrepancy between $f(x)$ and $g(x)$. And unbiased IS estimator $\hat{P}_{IS,fail}$ can be calculated as (6):

$$\hat{P}_{IS,fail} = \hat{P}_{IS}(Y \in S)$$

$$= \frac{1}{M}\sum_{j=1}^{M} w(x_j) I(x_j) \xrightarrow[M \to +\infty]{a.s.} P(Y \in S) \tag{6}$$

With a proper $g(x)$, $\hat{P}_{IS,fail}$ can be approximately equal to MC results.

The optimal sample distribution $g^{opt}(x)$ can be defined as:

$$g^{opt}(x) = \frac{I(x) \cdot f(x)}{P_{fail}} \tag{7}$$

However, it is infeasible to calculate $g^{opt}(x)$ with equation (7) analytically. Most of the works construct the sampling distribution by shifting the mean vector of original distribution to the boundary of failure region [8][9], or to the center of failure region [5]. However, these static IS methods with predefined $g(x)$ suffers from 2 drawbacks:

First, the efficiency of their methods is seriously affected by their initial conditions. They typically consist of two steps: searching the Optimal Shift Vector (OSV) and executing the Importance Sampling to estimate P_{fail}. Once the shifted vector determined, the efficiency of the whole algorithm has almost been determined. However, the cost of searching shift-vector increases greatly with dimension increases especially if the boundary has a complicated shape. One of solutions is updating estimator P_{fail} during searching the failure region in an adaptive scheme.

Moreover, some likelihood ratios become dominate especially when the dimension is high. The variance of likelihood ratio $w(x)$ is very large. Hence, the estimator shows huge numerical instability.

III. Proposed Method

A. Algorithm Overview

Algorithm 1 shows the entire flow of SSAIS. The objective of the entire algorithm is to dynamically adjust the position and shape of the distorted sampling distribution so that the sampling function is as close as possible to the true distribution within the failure region. First, we start from N multidimensional normal distribution $q_i^{(0)}(x) = q_i^{(0)}(x|u_i^{(0)}, \sum_i^{(0)})$ through the failure samples collected by hyperspherical pre-sampling, where the mean u_i denotes the location parameters, and the variance $\sum_i$ denotes the shape parameters. Meanwhile, we set a few batches (typically 2-3). Each batch contains a small number of iterations. Next, in each iteration, we construct a mixture sampling distribution $g^{(t-1)}$ from the N normal distribution $q_i^{(t)}$, i = 1, ..., N. Then we update the location parameters by resampling technique mentioned in [2]. After a batch of iterations, we reshape N normal distribution by changing the variance of the optimal sampling distribution mathematically. The failure rate will be also updated iteratively through an unbiased estimator.

Algorithm 1: SSAIS Algorithm

Initialization:

1. Set batch index $\mathbf{ba} = \mathbf{0}$ and a small number of iterations.
2. Construct N multinormal distribution through N failure samples:

$$\mathbf{q_i^{(0)}(x)} = \mathbf{q_i^{(0)}(x|u_i^{(0)}, \sum_i^{(0)})} \qquad \mathbf{i = 1, \dots, N}$$

where $\boldsymbol{u_i^{(0)} = x_i^{(0)}} \qquad \boldsymbol{i = 1, \dots, N}$,

$\boldsymbol{\Sigma_i^{(0)}}$ is normalized the unit matrix.

Repeat

Update batch index $\mathbf{ba} = \mathbf{ba} + \mathbf{1}$

For each iteration:

Update the location of Distort sampling distribution

1. Construct a Gaussian mixture distribution

$$\boldsymbol{g(x)^t = \frac{1}{N} \cdot \sum_{i=1}^{N} q_i^{(t)}\left(x \middle| u_i^{(t)}, \Sigma_i^{(t)}\right)} \qquad \boldsymbol{i = 1, \dots, N}$$

2. Update the unbiased estimator

(a) Regenerate N samples from $\boldsymbol{g(x)^t}$ and compute the weight for each sample:

$$\boldsymbol{w_{i,t} = \frac{\pi(x)}{g(x)^{(t)}} = \frac{f(x) \cdot I(x)}{g(x)^t}}$$

(b) Update the unbiased estimator:

$$\boldsymbol{\hat{P}_{fail}^t = \frac{1}{tN} \sum_{j=1}^{t} \sum_{i=1}^{N} w_{i,j}} \qquad \boldsymbol{i = 1, \dots, N}$$

3. Generate a new set of location parameters

(a) Normalize the weights for resampling:

$$\boldsymbol{\bar{w}_{i,t} = \frac{w_{i,t}}{\sum_{i=1}^{N} w_{i,t}}} \qquad \boldsymbol{i = 1, \dots, N}$$

(b) Update location parameters by resampling:

Replicate samples with large $\boldsymbol{\bar{w}_{i,t}}$ to increase the ratio of samples with higher importance. Update the location parameters by:

$$\boldsymbol{u_i^t = x_i^t} \qquad \boldsymbol{i = 1, \dots, N}$$

End

Reshape the distort sampling distribution

Update the shape parameters by EM algorithm derived from MLE:

(a) Calculate the "hidden variables":

$$\boldsymbol{r_k(x_i) = \frac{q_k(x_i)}{g(x_i)^t}} \qquad \mathbf{i = 1, \dots, N;} \quad \mathbf{k = 1, \dots, N}$$

where $\boldsymbol{r_k(x_i)}$ indicates the probability that $\boldsymbol{i}$**th** sample is from the *k*th component of $\mathbf{g(x)}$.

(b) Generate the optimal variance of sampling distribution:

$$\boldsymbol{\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} r_k(x_i)(x_i - u_k)}{\sum_{i=1}^{N} r_k(x_i)}}$$

Until

The Fig of merit for estimator: $\boldsymbol{\rho = \frac{\sqrt{\sigma_{\hat{P}_{fail}}^2}}{\hat{P}_{fail}} \leq 0.1}$

end

B. Initialization

One of the most important issues is how to collect suitable N failure samples to initialize the location and shape parameters of the first iteration. We use the spherical sampling to collect samples for it can explore the whole make the whole parametric space fully to keep the variety of samples. As a result, the whole algorithm can convergence more quickly.

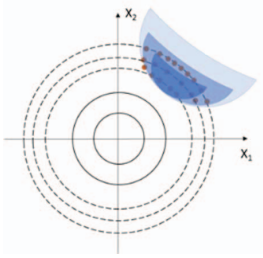

Fig 2. Collecting initial samples through Spherical Sampling

As shown in Fig 2, we first normalize all of the parameters as the unit hypersphere. Then we collect failure samples by increasing the radius gradually. The number of failure samples is a balance between efficiency and accuracy of the estimation. We stop the collection until more than 5% failure samples among 1000 samples are appeared in the current hypersurface, which means there are more than 100 failure samples with the ones in previous hypersurfaces. It is enough to guarantee estimation accuracy through our experiments. The different transparency areas in Fig 2 present different importance of failure samples which the samples fall in darker area have higher importance.

C. Construct Optimal Distort Sampling Distribution

C.1. Update Location Parameters

In order to make the current distribution $g(x)^t$ increasingly close to the true distribution $\pi(x)$ in the failure region, we first change the mean of each component in $g(x)^t$, that is the u_i^t of $q_i^{(t)}\left(x \middle| u_i^{(t)}, \Sigma_i^{(t)}\right)$, iteratively by the multidimensional resampling technique [2]. The basic idea is replicating the samples according to their importance measured by normalized weights, which is also a process of weight balance [2] to overcome the weight degeneracy suffered from traditional IS methods. As shown in Fig 3, after resampling, we shift $q_i^{(t)}\left(x \middle| u_i^{(t)}, \Sigma_i^{(t)}\right)$ to the position of these samples and regenerate the sample sets for next iteration. As the iteration continues, our sampling distribution $g(x)^t$ will be closer to the true distribution $\pi(x)$.

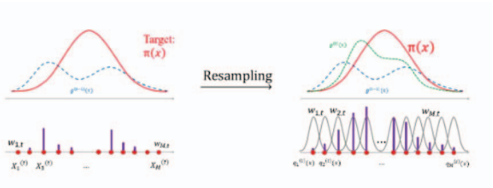

Fig 3. The process of approximating the true distribution in the failure region by resampling

C.2 Update Shape Parameters

Through the scheme mentioned above can enforce our sampling distribution $g(x)$ to be closer to real failure region distribution and result in a good estimation. However,

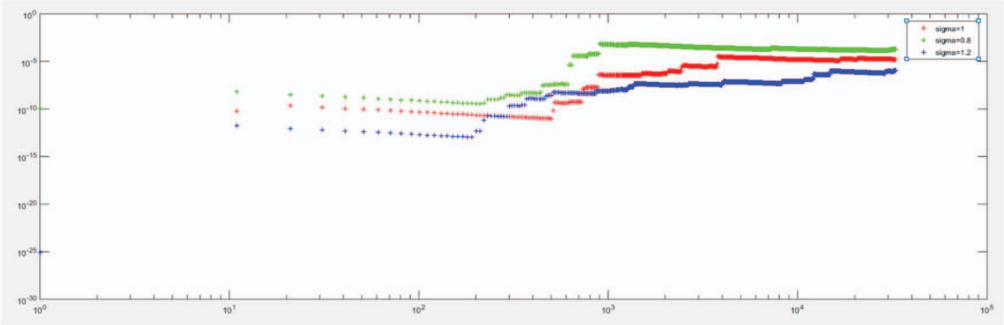

Fig 4. The relationship between sigma and convergence of the AIS algorithm

convergence speed is too slow. In our experiments, the whole algorithm spends at least ten thousand simulations (i.e, about one hundred iterations) to convergence a stable result when the variance has been set to 1σ, where the σ is the normalize device variation. In fact, it is what the AIS [2] did. Fig 4 shows the relationship between the variance and the efficiency of AIS within two million simulations (about 1000 iterations). We notice that the estimations show great differences with different variance. To be specific, the estimation with small variance has reached a reasonable estimation with the fastest speed. However, it convergences to a completed wrong estimation with 2.7e-4 while the standard MC result is 2.35e-5. It is because the sampling distribution $g(x)$ only concentrate a small of high probability failure regions while neglecting the other high probability regions. The diversity of samples regenerated by $g(x)$ is losing as the iteration continues so that it can't explore the parametric space fully. The estimation with large variance is about 3.5e-6 which is much smaller than the MC result. If we can further increase the simulation times, the estimation can convergence to the right value.

To address this issue, we adopted a strategy of dynamically adjusting variance in batches. To be specific, we divide the whole iteration into several batches. Each batch has a different variance calculated from the previous batch. To get the optimal variance, we used the EM algorithm derived from Maximum Likelihood Estimation (MLE). Notice that the $g(x)$ is actually a Gaussian mixture. For generality, we assume:

$$\mathrm{G(x|\theta)} = \sum_{j=1}^{N} \alpha_j N_j\left(x_j; u_j, \Sigma_j\right) \quad (8)$$

$$s.t. \quad \sum_{j=1}^{N} \alpha_j = 1 \quad (9)$$

where $N_j\left(x_j; u_j, \Sigma_j\right)$ is a multinormal distribution.

Based on the regenerated samples x_j $(j = 1, \ldots, N)$, we can formulate the log-likelihood function of MLE as:

$$\max_{\theta} \log(g(x|\theta)) = \max_{\theta} \sum_{k=1}^{N} log \sum_{j=1}^{N} \alpha_j N_j(x_j; u_j, \Sigma_j) \quad (10)$$

The parameters in (10) are estimated by EM algorithm which searches the optimal parameter estimation by performing expectation step and maximization step iteratively.

In our experiments, we first introduce a "hidden" random variable γ_{kj} to denote the probability the kth sample is from the jth component of $g(x)$, that is the posterior probability of α_j. Then, in E-step, we calculate the expectation of γ with samples of the last iteration:

$$\hat{r}_{kj} = \frac{\alpha_j N_j(x_k; u_j, \Sigma_j)}{\sum_{j=1}^{N} \alpha_j N_j(x_k; u_j, \Sigma_j)} \quad (11)$$

And in M-step, we update the optimal variance using optimal γ as:

$$\Sigma_j = \frac{\sum_{j=1}^{N} \gamma_{jk}(x_j - u_j^T)(x_j - u_j)}{\sum_{j=1}^{N} \gamma_{jk}} \quad (12)$$

To further decrease the cost of updating Σ, each batch only contains a small number of iterations to prevent diversity losing during the iterations with small variance and to improve the convergence speed of the whole algorithm without wasting simulations as much as possible.

D. Unbiasedness of SSAIS Estimator

To improve the tolerance to poor initialization, we update the failure probability during searching failure region iteratively by a developed estimator which guarantees the estimation built from final samples $\{x_i^{(t)}, w_{i,t}\}_{i=1}^{N}$ converge to the failure probability P_{fail}. We prove its unbiasedness with (13)~(16)

$$E[\hat{P}_{fail}^{t}] = E[\frac{1}{tN}\sum_{j=1}^{t}\sum_{i=1}^{N} w_{i,j}] \quad (13)$$

$$= \frac{1}{tN}\sum_{j=1}^{t}\sum_{i=1}^{N} E\left[\frac{f(x)I(x)}{g^j(x)}\right] \quad (14)$$

$$= \frac{1}{tN}\sum_{j=1}^{t}\sum_{i=1}^{N} \int \frac{f(x)I(x)}{g^j(x)} g(x)dx \quad (15)$$

$$= \frac{1}{tN}\sum_{j=1}^{t}\sum_{i=1}^{N} \int f(x)I(x)dx = P_{fail} \quad (16)$$

IV. Experimental Results

The proposed method SSAIS will be verified on a single 6-T SRAM with 54 variables cell firstly. And then we will validate SSAIS on a high-dimension sense amplifier with 63 variables. We also implement HDIS[7], AIS [2] for accuracy and efficiency comparison. The experiments are performed with MATLAB and HSPICE under 40nm SMIC model.

A. Experiments On 6T SRAM Cell

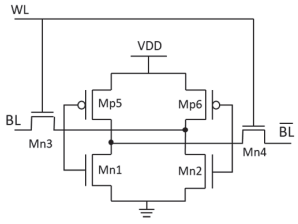

Fig 5. Circuits of a 6T SRAM cell

Table I. Accuracy and Efficiency Comparison on 54-dimensional SRAM cell

	MC	HDIS	AIS	SSAIS
Failure Prob.	2.35e-5 (0%)	2.60e-5(10.6%)	2.29e-5(2.6%)	2.33e-5(0.8%)
Pre-sampling (#sim)	0	7000	3000	3000
Importance Sampling (#sim)	10e7	19300	8720	3660
Total (#sim)	10e7	26300(380X)	11720(853X)	6660(1501X)

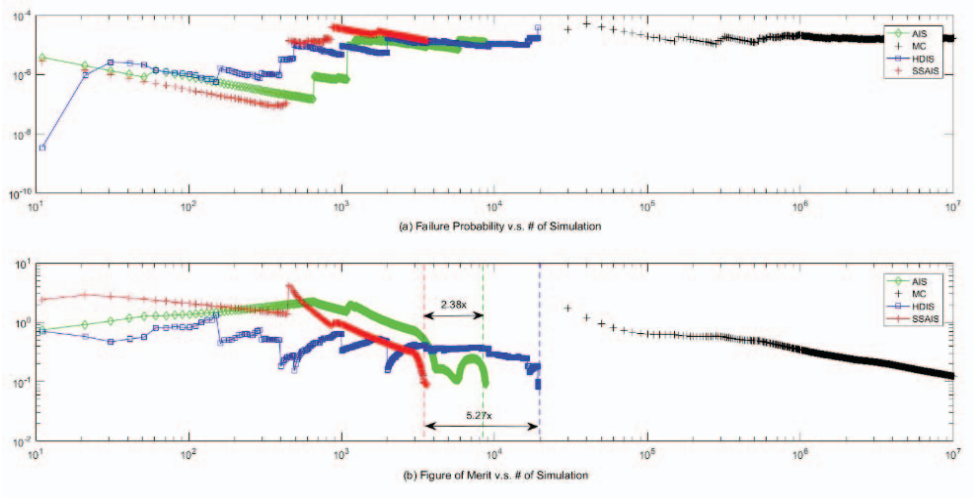

Fig 6. Evolution Comparison of Failure Prob. And FoM on SRAM cell

Fig. 5 shows the structure of a 6 transistor SRAM cell. The core of the cell is formed by two inverters consisted of four transistors: Mn1, Mn2, Mp5, Mp6, where the output of each inverter is fed as input into the other. This feedback loop stabilizes the inverters to their respective. The access transistors (Mn3, Mn4) and the word and bit lines, WL and BL, are used to read and write from or to the cell. We consider SRAM read voltage as the performance metric, which the failure event happens when the voltage difference between BL and $\overline{BL}$ at a given time is less than the threshold that the sense amplifier can sense.

A.1. Accuracy Comparison

We apply Fig of Merit (FOM) ρ to verify our proposed method, defined in [8][9]:

$$\rho = \frac{\sqrt{VAR_{\hat{P}_{fail}}}}{\hat{p}_{fail}} \tag{17}$$

Where the $VAR_{\hat{p}_{fail}}$ is the variance of $\hat{P}_{fail}$. And $\rho < \varepsilon\sqrt{log(1/\delta)}$ means one estimation has reached $(1-\varepsilon)100\%$ accuracy with $(1-\delta)100\%$ confidence. Here, we set ρ =0.0865 which means 95% accuracy with 95% confidence level.

We compare the convergence and FOM of SSAIS with other methods, as shown in Fig 6. From Fig 6 (b), we can see all methods have converged to a stable value with the FOM less than 0.0865. Table 1 illustrates the converged failure rate estimated by different methods. The MC result is 2.35e-5 as the ground truth. And HDIS, AIS, SSAIS have a relative error of 10.6%, 2.6%, 0.8% respectively. Note that the standard deviation of the distribution in AIS remains unchanged.

A.2. Efficiency Comparision

The efficiency of MC, HDIS, AIS and SSAIS are shown in Fig 6 b). We noticed that the proposed SSAIS has the fastest convergence to the 95% confidence interval within 6660 points, which speeds up 5.2X over HDIS and 2.38X over AIS for they have reached to 26300 points and 11720 points respectively, as illustrated in Table I. The better result of our proposed method can be attributed to search both the location and shape parameters dynamically, which is more efficient than traditional static IS.

B. Experiments On Sense Amplifier circuit

We make the experiment on Sense Amplifier which is one of the most important components in SRAM circuits.

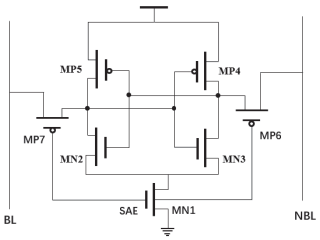

Fig 7. Schematic of Sense Amplifier

As shown in Fig 7, it mainly composes of two cross-coupled inverters that have up to 7 transistors. We consider the failure of SA that can't sense the voltage difference between BL and NBL due to process variation.

Table II. Accuracy and Efficiency Comparison on Sense Amplifier

	MC	HDIS	AIS	SSAIS
Failure Prob.	3.14e-5 (0%)	3.49e-5(11.2%)	3.23e-5(2.8%)	3.10e-5(1.2%)
Pre-sampling (#sim)	0	8000	3000	3000
Importance Sampling (#sim)	10e7	29650	7400	2520
Total (#sim)	10e7	37650(265X)	10400(961X)	5520(1811X)

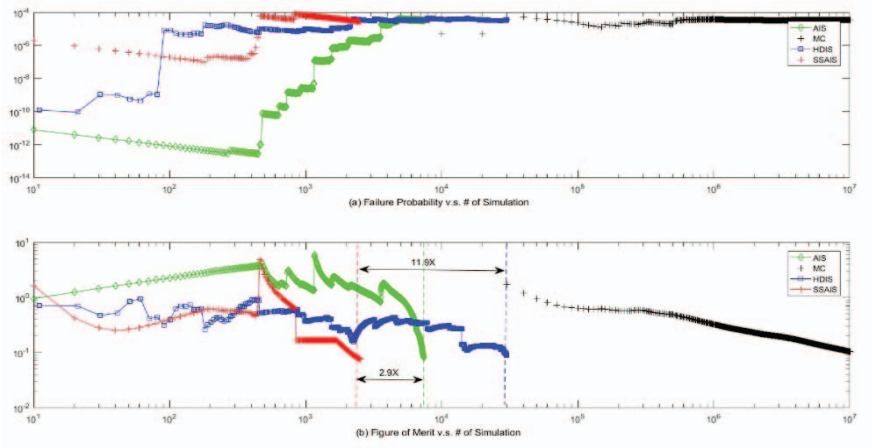

Fig 8. Evolution Comparison of Failure Prob. And FoM on Sense Amplifier

Accuracy and Efficiency Comparision

Fig 8 shows the convergence process of failure rate estimation and FOM in different methods. The MC result is 3.14e-5 evaluated with 10e7 simulations. And the proposed method has the faster speed to convergence to the reasonable failure probability (FoM<0.1) among the state-of-art methods. To be specific, as illustrated in Table II, SSAIS just take 5520 simulations with less than 2% relative error while HDIS and AIS need 37650 and 10400 simulations, respectively. Hence our method outperforms the other two methods about 2X~7X which gain 1811X speedup over MC. Note that samples used in IS iterations are the failure samples among 3000 samples collected in pre-sampling step. Hence it is possible the cost in IS step is less than that in pre-sampling.

V. Conclusion

In this paper, we present a Scaled-Sigma Adaptive Importance Sampling method to estimate the failure probability of SRAM components efficiently. The method is an extension of AIS which updates both location parameters and shape parameters by searching failure region iteratively. The experiment on SRAM cell shows that the proposed method achieves 1500X speedup over Monte Carlo and 2~5X over other state-of-art methods. Another experiment on sense amplifier indicates SSAIS outperforms Monte Carlo 1811X and the other method 2~11X.

VI. Acknowledge

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61874152.

References

[1] Chandrakasan, A. P., Bowhill, W. J., & Fox, F. (2000). Design of high-performance microprocessor circuits. Wiley-IEEE press.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Shi, Xiao, et al. "A fast and robust failure analysis of memory circuits using adaptive importance sampling method." Proceedings of the 55th Annual Design Automation Conference. ACM, 2018.

[3] Singhee, Amith, and Rob A. Rutenbar. "Statistical blockade: a novel method for very fast Monte Carlo simulation of rare circuit events, and its application." 2007 Design, Automation & Test in Europe Conference & Exhibition. IEEE, 2007.

[4] Singhee, Amith, et al. "Recursive statistical blockade: An enhanced technique for rare event simulation with application to SRAM circuit design." 21st International Conference on VLSI Design (VLSID 2008). IEEE, 2008.

[5] Wu, Wei, et al. "REscope: High-dimensional statistical circuit simulation towards full failure region coverage." Proceedings of the 51st Annual Design Automation Conference. ACM, 2014.

[6] Kanj, Rouwaida, Rajiv Joshi, and Sani Nassif. "Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events." 2006 43rd ACM/IEEE Design Automation Conference. IEEE, 2006.

[7] Wu, Wei, et al. "A fast and provably bounded failure analysis of memory circuits in high dimensions." 2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2014.

[8] Dolecek, Lara, et al. "Breaking the simulation barrier: SRAM evaluation through norm minimization." Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design. IEEE Press, 2008.

[9] Qazi, Masood, et al. "Loop flattening & spherical sampling: Highly efficient model reduction techniques for SRAM yield analysis." 2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010). IEEE, 2010.

[10] Wu, Wei, Srinivas Bodapati, and Lei He. "Hyperspherical clustering and sampling for rare event analysis with multiple failure region coverage." Proceedings of the 2016 on International Symposium on Physical Design. ACM, 2016.

[11] Bhavnagarwala, Azeez J., Xinghai Tang, and James D. Meindl. "The impact of intrinsic device fluctuations on CMOS SRAM cell stability." IEEE journal of Solid-state circuits 36.4 (2001): 658-665.

[12] Wang, Mengshuo, et al. "High-dimensional and multiple-failure-region importance sampling for SRAM yield analysis." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 25.3 (2016): 806-819.

[13] Katayama, Kentaro, et al. "Sequential importance sampling for low-probability and high-dimensional SRAM yield analysis." Proceedings of the International Conference on Computer-Aided Design. IEEE Press, 2010.