An Efficient SRAM Yield Analysis and Optimization Method With Adaptive Online Surrogate Modeling

Jian Yao, *Student Member, IEEE*, Zuochang Ye, *Member, IEEE*, and Yan Wang

***Abstract*— SRAM cells usually require extremely low failure rate or equivalently extremely high production yield, making it impractical to perform yield analysis using Monte Carlo (MC) method as huge amount of samples are needed. Fast MC methods, e.g., importance sampling methods, are still too expensive as the anticipated failure rate is very low. In this paper, a new SRAM yield analysis method is proposed to tackle this issue. The key idea is to improve traditional importance sampling method with an efficient online surrogate model. Experimental results show that the proposed yield analysis method achieves 5×–22× speedup over existing state-of-the-art techniques without sacrificing estimation accuracy. Sigma distribution and schmoo plot can be quickly generated by the proposed method, which is very useful for realistic applications. Based on the proposed yield analysis method, an efficient yield optimization method has been developed to further automate the SRAM cell design procedure where process variations can be fully considered. Experimental results show that a fully automatic yield optimization for SRAM cells can be done within only a few hours.**

***Index Terms*— Failure rate, importance sampling, optimization, process variations, SRAM, statistical analysis, surrogate model, yield.**

I. Introduction

INTEGRATED circuit design not only calls for optimized nominal design, but also requires high robustness and yield against process variations. With technology feature size scaling toward the physical limit, process variations become a growing concern in IC designs. This in general requires time-consuming statistical methods such as Monte Carlo (MC) method for yield analysis [2], [3].

To be worse, highly repeated cells, such as SRAM cells, usually require extremely low failure rate on the per cell basis to ensure a moderate yield for the whole chip. Consider a small-sized SRAM chip with 1 million bitcells and targeted yield of 50%. To achieve the targeted chip yield, the yield of each bitcell needs to achieve $\geq$99.9999% [4]. For practical SRAM designs, standard MC method requires a huge amount of samples (10^7–10^9) to achieve the wanted accuracy [5], with each sample corresponding to a SPICE simulation. The reason why so many samples are needed is that most of the samples fall into the acceptable region, and only an extremely small fraction of samples are in the failure region.

Accelerated MC methods are thus required. Most of existing acceleration methods are based on the importance sampling methods [5]–[7]. The key idea of importance sampling methods is to sample the process parameter space based on a distorted probability density function (pdf) to increase the probability of observing failures.

Importance sampling provides a framework to solve the computational barrier. However, the performance of importance sampling highly depends on its implementation details, especially on how to define the distorted pdf. In addition, even with importance sampling, the required samples are still in the order of several thousands to tens of thousands, thus the computational cost is still very high.

Furthermore, in the circuit design flows, the designers not only need to calculate the yield for a specific design, but also need an automatic yield optimization algorithm to find the optimal design with the best yield across the whole design space automatically. Such yield optimization algorithm can deal with process variations impacting circuit yield and performances in an integrated manner in order to avoid costly design iterations [8], [9].

In this paper, we propose a fast SRAM yield analysis method based on the importance sampling framework. The key idea is to adaptively replace circuit simulation with online-trained surrogate model. This surrogate model is used in both stages of importance sampling to reduce the computational cost. Sigma distribution and yield schmoo plot can be quickly generated by the proposed yield analysis method. Meanwhile, an efficient yield optimization method has been developed to further automate the SRAM cell design procedure with process variations fully considered.

The rest of this paper is organized as follows. In Section II, the relevant background materials will be presented. The proposed yield analysis method and optimization method will be presented in Sections III and IV, respectively. Next, experimental results are given in Section V. Finally, conclusions are given in the Section VI.

II. Background

Suppose $X = [x_1, x_2, \ldots, x_m]^T$ is an m-dimensional random variable modeling process variations and its joint

Manuscript received April 22, 2013; revised January 22, 2014 and May 5, 2014; accepted July 2, 2014. Date of publication July 29, 2014; date of current version June 23, 2015. This work was supported in part by the Major State Basic Research Development Program of China (973 Program) under Grants 2010CB327403 and 2011CBA00604 and in part by the National Natural Science Foundation of China under Grants 61176034 and 61106031.

The authors are with the Institute of Microelectronics, Tsinghua University, Beijing 100084, China (e-mail: jianyao.thu@gmail.com; zuochang@tsinghua.edu.cn; wangy46@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2014.2336851

pdf can be defined as $p(X)$. Such random variables include the variations of threshold voltage V_{th}, oxide thickness T_{ox}, and gate length L_{eff}. Typically, X is a multivariate normal distribution [5]. Without loss of generality, we further assume that the random variables $[x_1, x_2, \ldots, x_m]$ in the vector X are mutually independent and standard normal (i.e., with zero mean and unit variance)

$$p(X) = \prod_{i=1}^{m} \left[\frac{1}{\sqrt{2\pi}} \cdot \exp\left(-\frac{1}{2} x_i^2 \right) \right] \tag{1}$$

and any correlated and jointly normal random variables can be transformed to independent random variables in (1) by principal component analysis [10].

The failure rate P_f of an SRAM cell can be mathematically represented as

wrong?

$$P_f = \int_{\Omega} p(X) \cdot dX \tag{2}$$

where Ω denotes the failure region, i.e., the subset of the variation space where the performances of interest [e.g., static noise margin (SNM)] do not meet the specification.

Alternatively, the failure rate in (2) can be defined as

$$P_f = \int_{-\infty}^{\infty} I(X) \cdot p(X) \cdot dX \tag{3}$$

where $I(X)$ represents the indicator function

$$I(X) = \begin{cases} 1 & X \in \Omega \\ 0 & X \notin \Omega. \end{cases} \tag{4}$$

The failure rate P_f can be estimated by the standard MC analysis [11]. The key idea is to draw M random samples $\{\widetilde{X}_1, \widetilde{X}_2, \ldots, \widetilde{X}_M\}$ from $p(X)$, and then obtain the circuit performances $\{\widetilde{f}_1, \widetilde{f}_2, \ldots, \widetilde{f}_M\}$ by SPICE simulations to calculate indicator function $\{I(\widetilde{X}_1), I(\widetilde{X}_2), \ldots, I(\widetilde{X}_M)\}$. The failure rate can be calculated as

$$\widetilde{P}_f^{\text{MC}} = \frac{1}{M} \sum_{i=1}^{M} I(\widetilde{X}_i). \tag{5}$$

For SRAM cell design, the failure rate P_f in (3) is extremely small and most random samples drawn in MC analysis do not fall into the failure region Ω. Hence, the aforementioned MC method is extremely expensive, or even infeasible.

The importance sampling techniques are widely used to accelerate SRAM yield analysis [5]–[7]. It aims at directly generating a large number of random samples in the failure region using a distorted pdf $q(X)$. In this case, the failure rate can be expressed as

$$P_f = \int_{-\infty}^{\infty} \frac{I(X) \cdot p(X)}{q(X)} \cdot q(X) \cdot dX. \tag{6}$$

Typically, importance sampling is implemented in two steps. In the first step, the distorted pdf $q(X)$ should be generated.

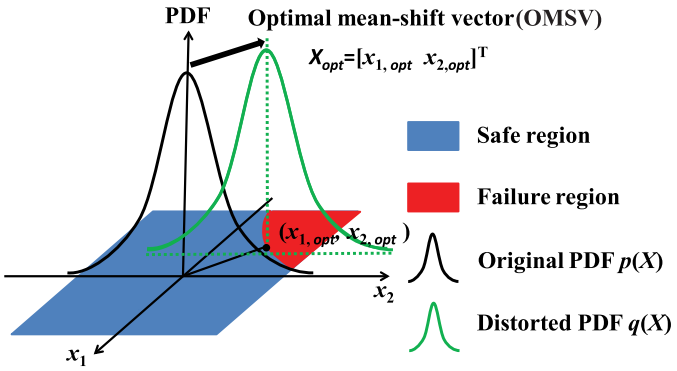

Fig. 1. 2-D example to illustrate the importance sampling.

Then, in the second step, N sampling points $\{\widetilde{X}_1, \widetilde{X}_2, \ldots, \widetilde{X}_N\}$ are drawn from $q(X)$ and the failure rate can be estimated by

$$\widetilde{P}_f^{\text{IS}} = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{I(\widetilde{X}_i) \cdot p(\widetilde{X}_i)}{q(\widetilde{X}_i)}. \tag{7}$$

The key of importance sampling methods is how to generate the distorted pdf $q(X)$. As discussed in [5], samples should be generated from the most likely failure region. Most existing importance sampling methods [5]–[7] are mainly based on this idea. For example, in [6], the maximum probability density of the process condition subject to violating at least one specification will be selected as the optimal mean-shift vector (OMSV). Then, the distorted pdf is based on this vector, which is illustrated in Fig. 1. In this 2-D example, the OMSV $X_{\text{opt}} = [x_{1,\text{opt}}, x_{2,\text{opt}}]^T$ is used to determine the most likely failure region and the distorted pdf $q(X)$ is based on it.

Although importance sampling technique reduces the computational cost in SRAM yield analysis, the samples (i.e., the SPICE-simulations) required in both steps of importance sampling are still expensive [5]–[7], and it needs to be further reduced.

III. Proposed Yield Analysis Method

The proposed yield analysis method follows the basic importance sampling framework introduced in Section II, i.e., to first find the OMSV and then, perform sampling based on the distorted pdf parameterized with OMSV.

A. Find OMSV With Optimization

According to (1), the maximum probability density occurs where the distance to the origin is minimized [12]. Thus, finding OMSV can be translated to solving the following global optimization problem:

$$\begin{aligned} \min \quad & \|X\| \\ \text{s.t:} \quad & X^- < X < X^+ \\ & I(X) = 0 \end{aligned} \tag{8}$$

where $\|\cdot\|$ is the two-norm.

The best class of algorithms for solving the above global optimization is the population-based optimization algorithm, which is illustrated in Fig. 2.

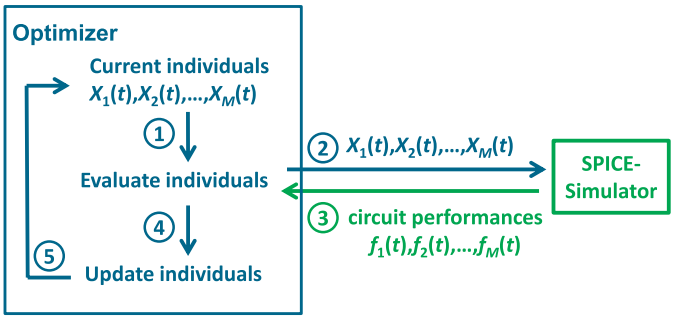

Fig. 2. Illustration of optimization to obtain the OMSV.

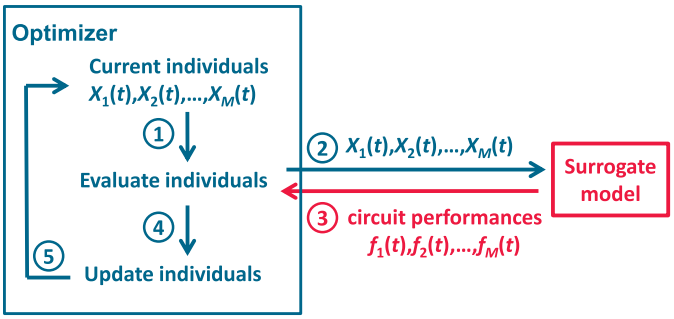

Fig. 3. Illustration of optimization combined with surrogate models.

At each generation (iteration) t, suppose $[X_1(t), X_2(t),...,X_M(t)]$ are the current individuals [Fig. 2(1)], where $X_i(t)$ is a sample in process parameter X, and M is the number of individuals in the population. Firstly, each individual will be evaluated by the SPICE-simulation [Fig. 2(2)] to calculate the circuit performance $f_i(t)$ $(i = 1, 2, \ldots, M)$ [Fig. 2(3)] and corresponding indicator function $I(X_i(t))$ $(i = 1, 2, \ldots, M)$ [Fig. 2(4)] according to (4). Then, the population will be updated [Fig. 2(5)] based on the adopted population-based optimization algorithm. The next iteration is then performed until the optimal individual (i.e., OMSV) satisfies the convergence condition.

In general, a lot of population-based optimization methods can be used, e.g., genetic algorithm [13] and particle swarm algorithm [14]. In this paper, we use the differential evolution algorithm [15] as the optimizer to solve this optimization problem.

The key problem of population-based optimization is that usually a large amount of individuals are required in order to obtain a reasonably good solution, and each individual needs one SPICE simulation in each iteration. As there are always several iterations to obtain OMSV, the total computational cost is still high.

B. Accelerating Solving Optimization Problem (8) With Offline Training Surrogate Model

The key to reduce the computational cost is to reduce the number of time-consuming SPICE simulations. To accelerate the optimization, a general idea is to first train surrogate models to approximate the mapping from process parameters to circuit performances (i.e., $X \rightarrow f$). Then, the optimization can be done with the generated surrogate models and no SPICE simulation is needed. The process is illustrated in Fig. 3. The surrogate model is generally in order of magnitude faster than SPICE simulation since the surrogate model can be thought as the black box only describing the relations between circuit performance and process variations.

There are quite a few existing surrogate modeling techniques, e.g., response surface modeling [16] and artificial neural network modeling [17], which can be used to generate the surrogate model. In this paper, we use radial basis function network [18] as the surrogate model to approximate the mapping $X \rightarrow f$. With several training samples obtained by SPICE-simulations, the radial basis function network model can be built.

Based on the surrogate model, the computational cost in optimization can be significantly reduced. However, there are still much time cost in generating offline surrogate model since each training sample for surrogate model needs one time of SPICE simulation. Reducing the training samples can reduce the computational cost, but at the cost of loosing model accuracy, and a common issue for offline surrogate model is that the balance between the accuracy and efficiency is usually poor.

Another issue of offline surrogate model is that the training samples are randomly distributed. This is inefficient, and as the purpose of optimization is to find the optimal individual, more training samples should be used in the region around OMSV.

C. Further Accelerating Solving Optimization (8) With Online Training Surrogate Model

The idea of online surrogate model is inspired by the observation that during the optimization, the requirement for model accuracy varies. The model accuracy should be high enough where it is close to OMSV X_{opt}, and it can be relatively low where it is far from X_{opt}. This procedure should be done in an adaptive way.

Before the optimization, a small number of training samples are generated to train an initial coarse surrogate model. To improve convergence, we use the latin-hypercube sampling method [11] with uniform distribution in the search space X to generate the initial training samples.

Then, each iteration of optimization begins with training the surrogate model as illustrated in Fig. 4. At each iteration t, the current individuals [Fig. 4(1)] will first be evaluated by the current surrogate model [Fig. 4(2) and (3)]. Among these individuals, the failure individual with minimum two-norm is selected as the current optimal individual $X_{\text{opt}}(t)$ [Fig. 4(4)] and this individual is evaluated by SPICE-simulation. Next, $X_{\text{opt}}(t)$ will be added to the training set and the surrogate model will be trained and renewed [Fig. 4(5)], and it will be used in the next iteration. At the same time, the optimizer will update the population based on the optimization algorithm [Fig. 4(6) and (7)].

Notice that there is only one SPICE simulation in each iteration (for the optimal individual in each iteration), the cost of the optimization can be highly reduced. Meanwhile, as only the current optimal individual in each iteration is used to train the surrogate model, the accuracy of the model will be

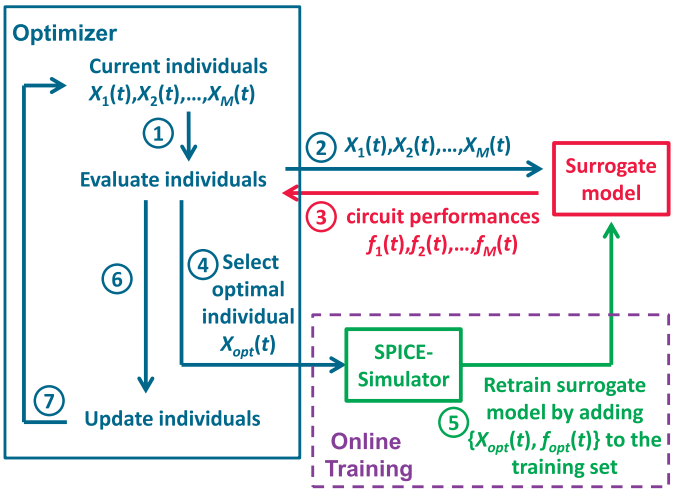

Fig. 4. Illustration of optimization with online training surrogate model.

sufficient in the region around OMSV. Therefore, the efficiency is achieved as most of training samples are used in the region of most interest.

D. Sampling From the Distorted pdf

When OMSV X_{opt} is generated by the optimization problem (8), the final trained surrogate model can also be generated at the same time based on the online training process. Then, the distorted pdf can be generated as follows to ensure minimum required samples [19]:

$$q(X) = \alpha p(X) + (1-\alpha)p(X - X_{\text{opt}}) \tag{9}$$

where $0 \leq \alpha \leq 1$.

For each sample, the indicator function $I(X)$ should be evaluated. In this method, we use the following strategy to calculate $I(X)$.

1) When the circuit performances predicted by the final trained surrogate model are far from the specification (e.g., SNM is far from 0), $I(X)$ will be evaluated from the surrogate model.
2) When the circuit performances predicted by the surrogate model are close to the specification, the SPICE simulation is done to obtain circuit performance and corresponding $I(X)$.

E. General Framework of the Yield Analysis Method

The overall algorithm of the proposed yield analysis method is shown in Algorithm 1.

IV. Proposed Yield Optimization Method

In practical SRAM designs, designers not only need to calculate the yield for a specific design, but also want to find the optimal design with the best yield in the design parameter space to avoid costly design iterations. Suppose that $Y = [y_1, y_2, \ldots, y_n]^T$ is an n-dimensional vector representing design parameters. Such random variables include parameters such as transistor widths and lengths, resistances, and capacitances. In general, the yield optimization problem can be

Algorithm 1 Framework of the Yield Analysis Method

Step 0: Generate the initial training set by Latin-hypercube sampling and train an initial surrogate model. Meanwhile, generate the initial optimization population.
Step 1: Calculate the circuit performance of current population based on the surrogate model.
Step 2: Add the current optimal individual to the training set and obtain the corresponding circuit performance by SPICE simulation.
Step 3: Retrain the surrogate model.
Step 4: Perform optimization to update the optimization population.
Step 5: Check whether the stopping criterion is met. If yes, obtain OMSV and final trained surrogate model, and go to step 6. Otherwise, go to step 1.
Step 6: Generate distorted PDF as described in equation (9).
Step 7: Sample from the distorted PDF using the strategy discussed in Section III-D.
Step 8: Obtain the failure rate and corresponding yield of SRAM cell by equation (7).

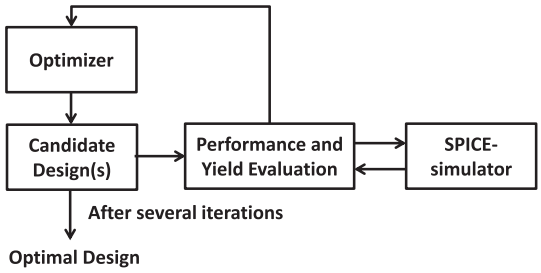

Fig. 5. General flow of yield optimization methods.

mathematically represented as the following yield optimization problem [9]:

$$\begin{aligned} &\max \quad \text{Yield}(Y) \\ &\text{s.t}: \quad Y^- < Y < Y^+ \end{aligned} \tag{10}$$

where the upper and lower bound $Y^{\pm}$ is determined by the technological process or the users setup.

The general yield optimization flow is summarized in Fig. 5 [9]. In the optimization loop, the candidate circuit designs are generated by the optimizer. Then, the performances and yield are analyzed based on several times of SPICE-simulation and fed back to the optimizer for the next iteration until the optimal circuit design satisfies the convergence condition.

To build an automatic yield optimization flow for SRAM design, a basic idea is that we can directly use the yield analysis method proposed in Section III to evaluate the yield and performances, and then the optimization work can be done based on the procedure shown in Fig. 5. However, the time cost of such optimization method is usually too high for practical circuit design. In this section, we propose a new yield optimization method to solve this problem.

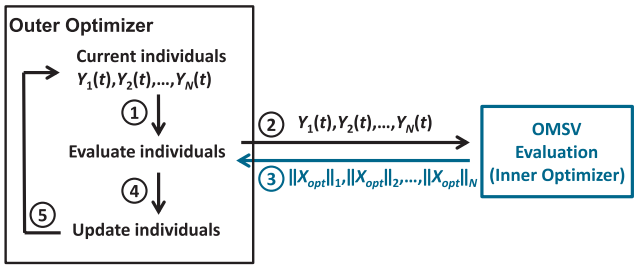

Fig. 6. Illustration the proposed yield optimization procedure.

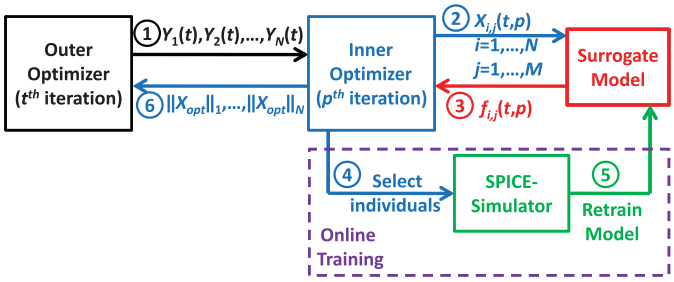

Fig. 7. Illustration the proposed yield optimization method with online training surrogate model.

A. Redefining the Optimization Objective

The norm of OMSV X_{opt} is a good indicator for estimating the probability of failure occurrence. In other words, finding the circuit design with the best yield can be approximated as finding the circuit design with the maximum two-norm of OMSV $\|X_{\text{opt}}\|$ [20]. So, in this paper, $\|X_{\text{opt}}\|$ is chosen as the optimization objective. According to (10), the yield optimization can be redefined as

$$\begin{aligned} &\max \quad \|X_{\text{opt}}\|(Y) \\ &\text{s.t:} \quad Y^- < Y < Y^+. \end{aligned} \tag{11}$$

Equation (11) is also a global optimization problem, and it can be solved by the population-based optimization algorithm as illustrated in Fig. 6. For the sake of discussion, (11) (for optimization) can be regarded as the outer optimization, and (8) (for finding the OMSV for each design) can be regarded as the inner optimization. At each generation (iteration), the two-norm of OMSV of each circuit design is calculated by inner optimizer [Fig. 6(2) and (3)]. Then, the population will be updated [Fig. 6(4) and (5)]. The next iteration is then performed until the optimal individual (i.e., the circuit design with the maximum $\|X_{\text{opt}}\|$) satisfies the convergence condition.

It must be noted that the maximum $\|X_{\text{opt}}\|$ is not exactly equal to the maximum yield. Thus, after the optimization problem has been solved, besides the optimal individual with maximum $\|X_{\text{opt}}\|$, a certain number of suboptimal individuals are preserved. Then, the yield of these individuals is calculated, and the circuit design with the maximum yield can be finally determined.

Compared with general yield optimization flow Fig. 5, the proposed optimization flow tries to optimize $\|X_{\text{opt}}\|$, which is much easier than optimizing the yield directly. This is the key to accelerate the procedure of yield optimization so that it can be used in practical applications.

B. Further Acceleration With Online Training Surrogate Model

To further reduce the time cost, the idea of online training surrogate model can be extended to the yield optimization flow after some necessary modifications.

As circuit design variables are involved in the yield optimization, the surrogate model should approximate not only the mappings from process variables to circuit performance ($X \rightarrow f$), but also the mappings from design variables to circuit performance ($Y \rightarrow f$). This problem can be easily solved by reorganizing independent variable as $V = [X, Y]$ so that the surrogate model can describe the mapping ($V \rightarrow f$) directly.

The flow of yield optimization with online training surrogate model is shown in Fig. 7. At each iteration (t, p) (where t and p are the current iteration of outer optimizer and inner optimizer, respectively), the current inner individuals can be expressed as $X_{i,j}(t, p)$ $(i = 1, 2, \ldots, N; j = 1, 2, \ldots, M)$, where N and M are the number of outer individuals and inner individuals, respectively. These individuals will be evaluated by current surrogate model [Fig. 7(2) and (3)]. Some of these will be selected [Fig. 7(4)] for SPICE simulation and added to the training set to renew the surrogate model [Fig. 7(5)]. After several iterations of inner optimizer [Fig. 7(2)–(5)], the OMSV can be determined and sent back to outer optimizer [Fig. 7(6)].

Selecting the individuals used to be added in the training sample set [Fig. 7(4)] is performed in two steps. First, for each circuit design, we select the failure individual with the minimum two-norm as the candidate samples $[X_{1,\text{opj}}(p),\ldots, X_{N,\text{opj}}(p)]$. In the second step, for these candidate samples $X_{i,\text{opj}}(p)$, we calculate the merit ρ

$$\rho = \frac{\|X_{i,\text{opj}}(p)\|}{\sigma_i(p)} \qquad (i = 1, 2, \ldots, N) \tag{12}$$

where $\sigma_i(p)$ is the standard deviation [6] for ith circuit design. The candidate samples with higher merit ρ will be selected, whereas those with lower merit will not be selected.

The above selection strategy is obvious, as larger $\|X_{i,\text{opj}}(p)\|$ generally means lower failure probability, and smaller $\sigma_i(p)$ indicates better accuracy of estimation. The failed individuals with the maximum two-norm and best estimation accuracy are the critical individuals, and they should be allocated more SPICE simulations.

C. General Framework of the Yield Optimization Method

The overall algorithm of the proposed yield optimization method can be described in Algorithm 2.

V. Experimental Results

In this section, the proposed method is verified by standard 6-T SRAM in a commercial 40-nm CMOS process. Fig. 8 shows the schematic view of the SRAM cell. The local V_{th} mismatch of each transistor is considered as the

Algorithm 2 Framework of the Yield Optimization Method

0: Train initial model. Generate initial outer population.
1: For each circuit design, generate initial inner population.
2: Current inner population are evaluated by current model.
3: Select individuals from current inner population based on the aforementioned strategy and do SPICE-simulation. Add these individuals to the training set. Retrain the model.
4: For each circuit design, perform inner optimization to update the inner population.
5: For each circuit design, check whether the stopping criterion is met. If yes for all designs, go to **6**. Otherwise, go to **2**.
6: Perform outer optimization to update outer population.
7: For outer optimizer, check whether the stopping criterion is met. If yes, obtain a certain number of optimal and sub-optimal solutions, and go to **8**. Otherwise, go to **1**.
8: Generate distorted PDF for these selected designs as described in equation (9).
9: For these designs, do distorted sampling to calculate the yield according to equation (7). Obtain the circuit design with the maximum yield as the optimization result.

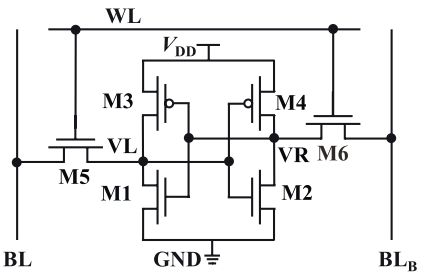

Fig. 8. Circuit schematic of the 6-T SRAM cell.

TABLE I
STATIC AND DYNAMIC METRICS OF SRAM

Failure Type	Static Metric	Dynamic Metric
Access Failure	Static Read Current (I_{read})	Read Access
Read Failure	Static Noise Margin (SNM)	Read Stability
Write Failure	Write Noise Margin (WNM)	Writeability

process variable. All process variables are mutually independent and standard normal.

A robust SRAM design should satisfy multiple specification requirements. Both static and dynamic metrics can be used to characterize the stability of the SRAM, which is shown in Table I. More details about metrics can be found in [21]–[23].

Two kinds of failure rate will be calculated.

1) *Failure Rate of Single Metric:* The failure rate of each single metric will be calculated individually, and they can be used to verify that the proposed method is valid for each metric (both static and dynamic).
2) *Failure Rate of Multiple Metrics:* All of metrics will be evaluated by the proposed method at the same time. If any metric of interest does not meet the specification, the circuit will be counted as a failure.

In Section V-A, we will verify the accuracy and efficiency of the proposed yield analysis method by calculating the failure rate for each metric. Then, more applications of the yield analysis method will be discussed in Section V-B. Finally, the yield optimization method will be verified in Section V-C, where the failure rate of multiple metrics will be considered.

A. Verification of the Yield Analysis Method

1) Static Metric Verification: Four specific SRAM designs are used as the experimental examples, named $\{\text{SRAM}_1, \text{SRAM}_2, \text{SRAM}_3, \text{SRAM}_4\}$, respectively. Read Noise Margin is selected as the static metric [24]. Both proposed method and standard MC method [11] are used to calculate the failure rate, which is shown in Fig. 9(a). The failure rate estimations from the two methods closely match each other, which validates the estimation accuracy of our proposed method. From Fig. 9(b), we can find that the required samples of the proposed method are 1e2×–1e5× less than the standard MC method.

To show the efficiency of the proposed method, yield analysis for SRAM cells is done with the proposed method, mixture importance sampling (MIS) [7], minimum-norm importance sampling (MNIS) [6], and Gibbs sampling (GS) [5]. The convergence condition is selected as the figure-of-merit (FOM) [6] equals to 0.1, corresponding to 90% accuracy level with 90% confidence interval.

Table II shows the computational cost of MIS, MNIS, GS, and the proposed method for SRAM_4. An 5×–22× accelerating rate of the proposed method compared with other exiting methods can be found, and similar accelerating rates are also shown in other SRAM examples $\{\text{SRAM}_1, \text{SRAM}_2, \text{SRAM}_3\}$. We can find that although the required samples for the proposed method are similar or even more than other exiting methods, most of the samples can be generated from the surrogate model in both steps of the importance sampling, which is much faster than SPICE simulation, and this greatly improves the efficiency.

2) Dynamic Metric Verification: Read Access, Read Stability, and Writeability are selected as the dynamic metrics. Similar to static metrics, dynamic metrics are also continuous scalar metric, which can be used in surrogate modeling. Therefore, the proposed method is also valid for dynamic metrics. The only difference for static and dynamic metrics is that dynamic metrics need transient simulation.

The failure rate of dynamic metrics is evaluated by the proposed method and standard MC method, respectively. The result given in Fig. 10 shows a good agreement between the proposed method and MC method for all dynamic metrics. The number of samples required for the proposed method is smaller than 5000, whereas MC method needs 10^6–10^7 samples to obtain the accurate results. The computational cost can be greatly reduced.

B. Application of the Yield Analysis Method

In this section, the proposed method is used to generate sigma distribution and yield schmoo plot, which is very useful in realistic SRAM design.

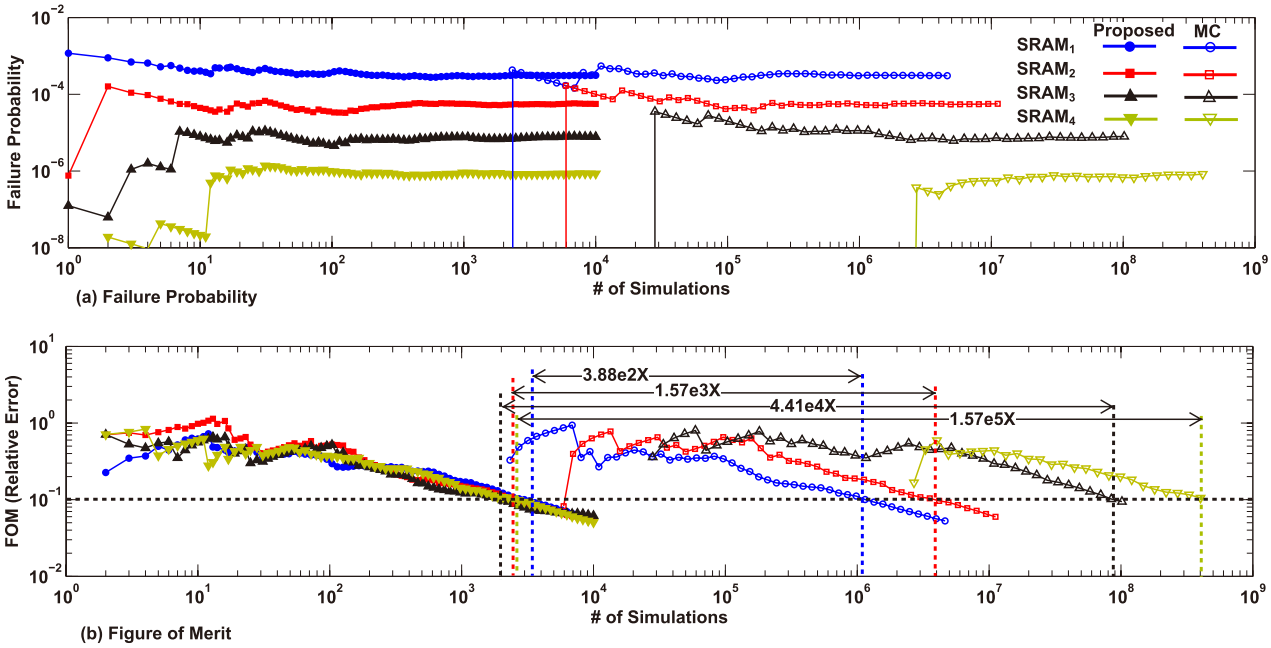

Fig. 9. Estimated failure rate and FOM as a function of the number of samples by the proposed method (the second step) and standard MC method for the four SRAM examples.

TABLE II
COMPUTATIONAL EFFICIENCY OF THE FOUR METHODS IN $SRAM_4$

Method	The first step			The second step			Totally cost	Speedup
	Samples	SPICE cost	Surrogate model cost	Samples	SPICE cost	Surrogate model cost		
MIS [7]	6600	6600×1.35s	0	7465	4740×1.35s	0	255.15min	1X
MNIS [6]	2400	2400×1.35s	0	2501	1588×1.35s	0	89.73min	2.84X
GS [5]	2385	2385×1.35s	0	584	371×1.35s	0	62.01min	4.11X
Proposed	2800	325×1.35s	2475×8.24ms	2473	159×1.35s	2317×8.24ms	11.55min	22.09X

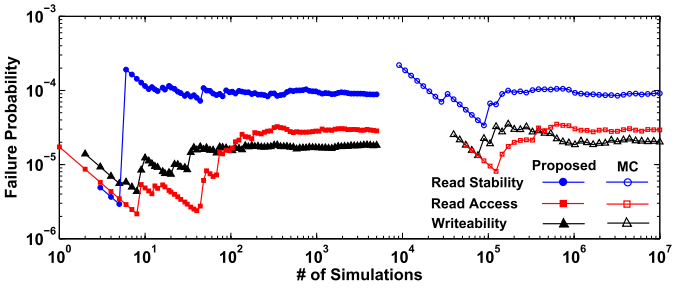

Fig. 10. Estimated failure rate of dynamic metrics as a function of the number of samples by the proposed method and standard MC method.

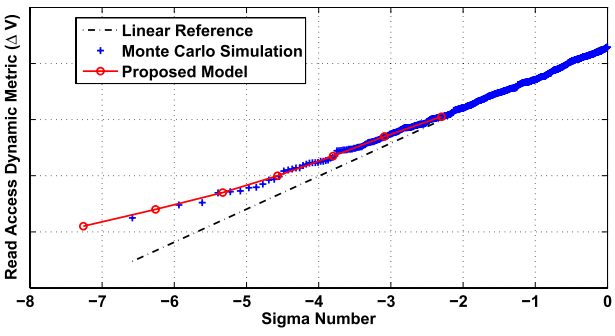

Fig. 11. Comparison of the proposed method and MC method in generating sigma distribution of Read Access dynamic metric.

1) Sigma Distribution Generation: The failure rate of Read Access metric corresponding to different value of Read Access specification is computed by the proposed method and MC method. Both results (in the form of sigma distribution) are shown in Fig. 11. We can find that the results from both methods match well with each other, verifying the accuracy of the proposed method. The time cost of the proposed method is ∼30 min, which is much faster than MC method.

2) Yield Schmoo Generation: The proposed method is used to generate the yield schmoo plots [25] against power supply voltages. The results are shown in Figs. 12 and 13, in which Vdd_wl&bl stands for the supply voltage of word line and bit line (WL/BL), and Vdd_cell stands for the supply voltage of the transistor. They all change from 0.85 to 1.25 V with step 0.05 V. The darkness of color represents the level of production yield.

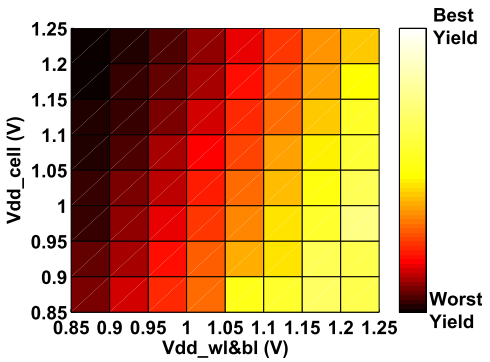

Fig. 12. Yield schmoo plot considering the failure rate of WNM metric.

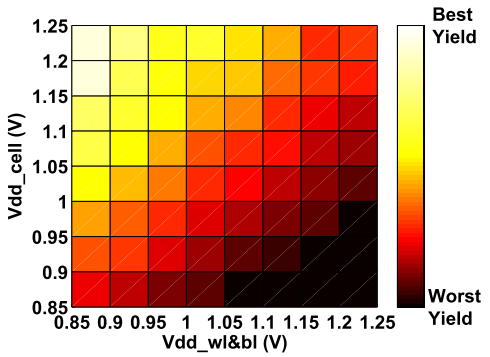

Fig. 13. Yield schmoo plot considering the failure rate of SNM metric.

Fig. 12 is the yield schmoo plot for WNM metric. The worst yield is found on the top left region (i.e., the maximum cell voltage supply and minimum WL/BL voltage supply). It agrees with common sense that when WL/BL voltage is low, it is hard to write into the bit cell and thus the WNM-referred yield is low, otherwise the WNM-referred yield is high. Fig. 13 is the yield schmoo plot for SNM metric, and it shows an opposite trend that also agrees with common sense that high WL/BL leads to read difficulty and thus the SNM-referred yield is low. In practice, the best overall yield is obtained by considering both schmoo plots and it should be along the diagonal of the plot.

C. Verification of the Yield Optimization Method

In this section, yield optimization is done considering six design parameters, including the widths and lengths of the drive, access, and load transistors. The transistor width ranges from 80 to 500 nm and the length ranges from 40 to 80 nm, which is the common setting in SRAM design [26]. Note that the same algorithm can be applied to other design parameters (e.g., the threshold voltage of each transistor). The failure rate will be calculated based on all of static metrics. The accuracy reference is based on traditional yield optimization flow, where the importance sampling [5] directly substitutes MC sampling to calculate the yield without other modification. All experiments are executed five times with different random starting points to verify the robustness of the proposed algorithm.

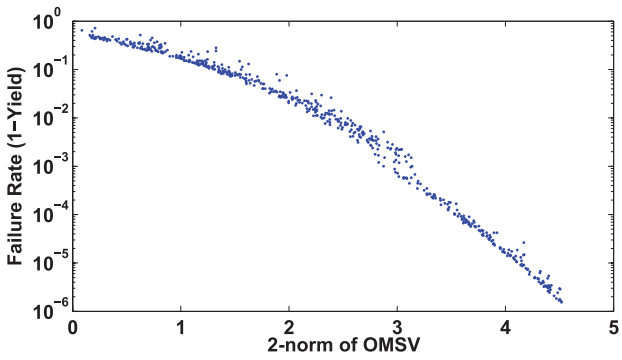

Fig. 14. Two-norm of OMSV and corresponding failure rate for 500 designs.

TABLE III
COMPUTATIONAL COST IN YIELD OPTIMIZATION

Method	Total Samples	SPICE Cost Sample $\times T_S$	Model Cost Sample $\times T_M$	Total Cost
Proposed	368371	10218*1.44s	358153*12.7ms	5.35 Hours
Reference	2022095	2022095*1.44s	-	33.7 Days

Firstly, to show that yield optimization can be approximated as the optimization of two-norm of OMSV $\|X_{\text{opt}}\|$, we randomly select 500 circuit designs with different $\|X_{\text{opt}}\|$, and calculate the corresponding failure rate (i.e., the yield), which is shown in Fig. 14. We can find that failure rate generally reduces with the increasing of $\|X_{\text{opt}}\|$. As we will evaluate the yield of several promising designs but not the single design with the largest $\|X_{\text{opt}}\|$, the error can be further reduced.

After using the proposed method and reference method for SRAM, the failure rates of the optimal design are 1.58E-6 (proposed method) and 1.73E-6 (reference method), respectively. The failure rates for both methods are close to each other.

The computational cost is shown in Table III. The required samples of the proposed method are smaller than the reference method. This is because the proposed flow optimizes the two-norm of OMSV $\|X_{\text{opt}}\|$, and the reference method needs to calculate the yield directly. Meanwhile, most of samples can be generated from the surrogate model, which is orders of magnitude faster than SPICE-simulation. The proposed method can solve the overall yield optimization problem in 5.35 h, whereas the reference method needs 33.7 days.

VI. CONCLUSION

In this paper, we proposed an efficient SRAM yield analysis and optimization method. The key idea of the proposed method is to accelerate the analysis by a novel online surrogate model to highly reduce the number of SPICE simulations. Population-based optimization algorithm is used as the core optimizer. Experimental results show that the proposed yield analysis method achieves 5×–22× speedup over existing state-of-the-art techniques without sacrificing estimation accuracy, and the proposed yield optimization method can optimize the design robustly within a few hours automatically.

Meanwhile, based on the proposed yield analysis method, the sigma distribution and yield schmoo can be generated efficiently, which is very useful for realistic SRAM design.

REFERENCES

[1] J. Yao, Z. Ye, and Y. Wang, "Efficient importance sampling for high-sigma yield analysis with adaptive online surrogate modeling," in *Proc. Conf. Design Autom., Test Eur. (DATE)*, Mar. 2013, pp. 1291–1296.

[2] J. Yao *et al.*, "Statistical analysis of soft error rate in digital logic design including process variations," *IEEE Trans. Nucl. Sci.*, vol. 59, no. 6, pp. 2811–2817, Dec. 2012.

[3] J. Yao, Z. Ye, and Y. Wang, "Statistical analysis of process variations in RF/mm-wave circuits with on-the-fly passive macro-modeling," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 2, pp. 727–735, Feb. 2013.

[4] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*. Philadelphia, PA, USA: SIAM, 1992.

[5] C. Dong and X. Li, "Efficient SRAM failure rate prediction via Gibbs sampling," in *Proc. 48th Des. Autom. Conf. (DAC)*, Jun. 2011, pp. 200–205.

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

[7] 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. 43rd ACM/IEEE Design Autom. Conf.*, Jun. 2006, pp. 69–72.

[8] M. Bühler *et al.*, "DFM/DFY design for manufacturability and yield—Influence of process variations in digital, analog and mixed-signal circuit design," in *Proc. Conf. Design Autom., Test Eur. (DATE)*, vol. 1. Mar. 2006, pp. 1–6.

[9] B. Liu, F. V. Fernandez, and G. G. E. Gielen, "Efficient and accurate statistical analog yield optimization and variation-aware circuit sizing based on computational intelligence techniques," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 6, pp. 793–805, Jun. 2011.

[10] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes with Errata Sheet*. New York, NY, USA: McGraw-Hill, 2001.

[11] X. Li, J. Le, and L. T. Pileggi, "Statistical performance modeling and optimization," *Found. Trends Electron. Des. Autom.*, vol. 1, no. 4, pp. 331–480, Apr. 2006.

[12] T. McConaghy and P. Drennan, "Variation-aware custom IC design: Improving PVT and Monte Carlo analysis for design performance and parametric yield," in *Solido White Paper*. San Jose, CA, USA: Solido Design Automation, Inc., 2011.

[13] K. F. Man, K. S. Tang, and S. Kwong, *Genetic Algorithms: Concepts and Designs*. New York, NY, USA: Springer-Verlag, 1999.

[14] J. Sun, C.-H. Lai, and X.-J. Wu, *Particle Swarm Optimisation: Classical and Quantum Perspectives*. Boca Raton, FL, USA: CRC Press, 2011.

[15] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution. A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag, 2005.

[16] A. I. Khuri, *Response Surface Methodology And Related Topics*. Singapore: World Scientific, 2006.

[17] B. Yegnanarayana, *Artificial Neural Networks*. New Delhi, India: PHI Learning Pvt. Ltd., 2004.

[18] S.-F. Su, C.-C. Chuang, C. W. Tao, J.-T. Jeng, and C.-C. Hsiao, "Radial basis function networks with linear interval regression weights for symbolic interval data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 1, pp. 69–80, Feb. 2012.

[19] A. Singhee and R. A. Rutenbar, "Statistical blockade: A novel method for very fast Monte Carlo simulation of rare circuit events, and its application," in *Proc. Conf. Design Autom., Test Eur. (DATE)*, Apr. 2007, pp. 1–6.

[20] H. E. Graeb, *Analog Design Centering and Sizing*. Heidelberg, Germany: Springer-Verlag, 2007.

[21] B. Zimmer *et al.*, "SRAM assist techniques for operation in a wide voltage range in 28-nm CMOS," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 12, pp. 853–857, Dec. 2012.

[22] S. O. Toh, Z. Guo, T.-J. K. Liu, and B. Nikolic, "Characterization of dynamic SRAM stability in 45 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 46, no. 11, pp. 2702–2712, Nov. 2011.

[23] D. E. Khalil, M. Khellah, N.-S. Kim, Y. Ismail, T. Karnik, and V. K. De, "Accurate estimation of SRAM dynamic stability," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 12, pp. 1639–1647, Dec. 2008.

[24] K. Agarwal and S. Nassif, "The impact of random device variation on SRAM cell stability in sub-90-nm CMOS technologies," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 1, pp. 86–97, Jan. 2008.

[25] R. Kanj, R. Joshi, C. Adams, J. Warnock, and S. Nassif, "An elegant hardware-corroborated statistical repair and test methodology for conquering aging effects," in *IEEE/ACM Int. Conf. Comput.-Aided Design, Dig. Tech. Papers*, Nov. 2009, pp. 497–504.

[26] V. Gupta and M. Anis, "Statistical design of the 6T SRAM bit cell," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 1, pp. 93–104, Jan. 2010.

Jian Yao (S'12) received the B.S. degree in electronic engineering from Harbin Institute of Technology, Harbin, China, in 2010. He is currently pursuing the Ph.D. degree at the Institute of Microelectronics, Tsinghua University, Beijing, China.

He has been a Visiting Scholar with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA, USA, since 2013. His current research interests include computer-aided design for integrated circuits, in particular, statistical analysis, device modeling, and yield optimization.

Zuochang Ye (M'08) received the B.S. and Ph.D. degrees from Tsinghua University, Beijing China, in 2002 and 2007, respectively.

He was a Research Scientist with Cadence Research Laboratories, Berkeley, CA, USA, from 2007 to 2008. He is currently an Associate Professor with the Institute of Microelectronics, Tsinghua University. His current research interests include computer-aided design for very large-scale integration circuits, in particular, numerical algorithms for electromagnetic simulation and circuit simulation.

Yan Wang received the Ph.D. degree in semiconductor device and physics from the Chinese Academy of Sciences, Beijing, China, in 1995.

She has been a Professor with the Institute of Microelectronics, Tsinghua University, Beijing, since 1999.