PAPER *Special Section on VLSI Design and CAD Algorithms*

Efficient Aging-Aware Failure Probability Estimation Using Augmented Reliability and Subset Simulation

Hiromitsu AWANO[†a)] ***and*** **Takashi SATO**[††], ***Members***

SUMMARY A circuit-aging simulation that efficiently calculates temporal change of rare circuit-failure probability is proposed. While conventional methods required a long computational time due to the necessity of conducting separate calculations of failure probability at each device age, the proposed Monte Carlo based method requires to run only a single set of simulation. By applying the augmented reliability and subset simulation framework, the change of failure probability along the lifetime of the device can be evaluated through the analysis of the Monte Carlo samples. Combined with the two-step sample generation technique, the proposed method reduces the computational time to about 1/6 of that of the conventional method while maintaining a sufficient estimation accuracy.

key words: *failure probability calculation, NBTI, Monte Carlo, subset simulation, augmented reliability*

1. Introduction

Aggressive scaling of a semiconductor manufacturing technology has enabled us to integrate billions of transistors into a single silicon chip. The increasing manufacturing variability, on the other hand, requires us to consider various uncertainty sources making the design of reliable circuits a significantly challenging task. Representative examples can be found in designs of an SRAM bit-cell or a D Flip-Flop (DFF). Considering that a modern large-scale integration (LSI) circuit embeds millions of such components, each circuit component needs to be highly resilient to the manufacturing variability to achieve satisfiable yield. For example, the failure probability of an SRAM cell must be extremely low — as low as 10^{-8} to 10^{-6}, or below [1]. Simulation methods that can efficiently estimate the rare circuit failure probability are, therefore, becoming increasingly important in circuit design.

The rare event simulation is known to be a difficult task. A straightforward approach for realizing such failure probability estimation is to rely on the naive Monte Carlo (MC) method, in which random samples are drawn from the statistical distribution corresponding to a process variability. Circuit simulations are then repeatedly conducted with those samples. However, since the target failure event is extremely rare, one may obtain very few failure samples even after utilizing millions of samples. The small number of failure samples makes the failure probability estimation inaccurate, or often makes it impossible when no failure sample is available. To resolve this issue, various methods have been proposed. Representative examples include well-known importance-sampling techniques [2], [3]. In the importance sampling, the original distribution is "distorted" so that more failure samples can be obtained, which drastically increases the estimation efficiency. However, the importance-sampling based methods still required extensive computational efforts to identify the suitable shape of the distorted distribution. The computational cost of the importance sampling tends to grow exponentially as the number of transistors, i.e., the number of uncertain variables in the circuit, increases. With the invention of subset simulation (SubSim) [4], failure probability estimations of larger circuits, such as DFFs or an SRAM array, became possible with a practical time limit [5].

Recently, failure probability estimation has become even more complicated than before — device aging has to be simultaneously dealt with. Transistors of extremely small channel lengths and increasing operating frequencies have accelerated device aging. As a result, device aging has emerged as not only an academic research topic but also an industrially important concern. The circuit failure probability provides vital information for designers to determine the critical parameters of the circuit, such as operational voltages or clock periods, and hence the estimation of the temporal change of the failure probability over the circuit-lifetime is becoming particularly important.

In [6] and [7], authors proposed a failure probability calculation that can take both initial process variability and device degradation into account. The impact of NBTI on the stability of an SRAM cell is analyzed in [8]. Those methods, however, rely on simplified models to make the calculations tractable. In the cases of [6] and [7], authors employed a response surface model to approximate inherently nonlinear relationships between the variability and corresponding circuit metrics, such as noise margin. In [8], noise margin of an SRAM cell is assumed to follow a normal distribution. However, approximation using the normal distribution is accurate only about the mean of the target distribution. The approximation accuracy is severely deteriorated at the rightmost tail of the distribution, which is the most important region in failure probability estimation. In order to avoid such approximations and to obtain an accurate failure probability estimation, an MC-based method is definitely required. An extension of the MC-based method that utilizes a

Manuscript received March 9, 2017.

Manuscript revised July 10, 2017.

[†] The author is with the VLSI Design and Education Center, The University of Tokyo, Tokyo, 113-8656 Japan.

[††] The author is with Graduate School of Informatics, Kyoto University, Kyoto-shi, 606-8501 Japan.

a) E-mail: awano@vdec.u-tokyo.ac.jp

DOI: 10.1587/transfun.E100.A.2807

Latin hyper-cube sampling is proposed in [9]. However, this method only aims to efficiently capture the temporal change of the outline of a performance distribution, so it is not suitable to apply for calculating rare circuit failure probability. To the best of our knowledge, there is no efficient method that gives the temporal change of circuit failure probability.

In this paper, we propose a method that simultaneously but efficiently handles device manufacturing variation and device aging to accurately calculate the temporal change of the rare circuit failure probability. For this purpose, a novel method that combines an augmented reliability approach and SubSim was developed. Hereafter, we call the augmented reliability extension of SubSim as SubSim-AR. SubSim-AR is originally proposed in [10] in order to assess the impact of structural design on its reliability. With SubSim-AR, the temporal change in rare failure probability can be estimated with only a single SubSim run. As compared to the conventional assessments through multiple simulation runs, SubSim-AR can reduce simulation time.

The contribution of our work is to apply and extend SubSim-AR for realizing the aging-aware failure probability calculation of the circuits. The key idea of our proposal is to treat the operation time (circuit age) of an LSI as a design parameter, enabling the analysis of failure probabilities over the course of the operation time with only a single SubSim run. However, since the device aging drastically changes the circuit failure probability, sufficient accuracy cannot be obtained by simply applying SubSim-AR for the circuit analysis. With this reason, a two-stage algorithm has been developed; in the first stage, the temporal change of the failure probability is coarsely estimated to calibrate parameters for SubSim-AR, and in the second stage, accuracy is improved with the calibrated parameters. The application of this method is not limited to the aging-aware circuit analysis. Wherever the multiple failure probability calculations are conducted to evaluate circuit reliability, the proposed method can be suitably applied for accelerating the simulations.

The rest of this paper is constructed as follows. In Sect. 2, background knowledge that forms the basis of our method is reviewed. The proposed method and its implementation will be described in Sect. 3. Then, in Sect. 4, results of numerical experiments are presented. Finally, conclusion is drawn in Sect. 5.

2. Preliminaries

2.1 Device Aging

In this section, a model for negative bias temperature instability (NBTI) is briefly reviewed as an example aging mode of a transistor. However, our approach is independent of aging modes; other aging modes, such as positive bias temperature instability (PBTI), hot carrier injection (HCI), and time dependent dielectric breakdown (TDDB), can be included similarly to or simultaneously with the case of NBTI.

NBTI is a phenomenon that the threshold voltage (V_{TH}) of a pMOS transistor increases gradually when negative bias is applied to its gate terminal. The degraded V_{TH} recovers, but often not perfectly, as soon as the negative bias is removed. This makes the modeling and the prediction of the impact of NBTI a very challenging task.

There are many models that are intended to predict the V_{TH} change due to NBTI [11]–[13]. Considering that our final goal is to calculate how the failure probability changes over a long period of operation time, we employ a long term prediction model [13], in which the magnitude of V_{TH} shift (ΔV_{TH}) is expressed as

$$\Delta V_{\mathrm{TH}} = k \cdot t^{n}. \tag{1}$$

Here, k is a model parameter that reflects the stress condition, operational temperature, and the chip fabrication process. t is an age of the transistor. n is also a model parameter which corresponds to the degradation speed. For scaled transistors, variability in device degradation is frequently observed [14]. In order to take this effect into consideration, we treat n as a random variable [13], i.e., the value of n varies from transistor to transistor. It is experimentally found that the statistical distribution of n can be well approximated by a log-normal distribution [15]. Hence, in this paper, we employ a log-normal distribution to replicate the variability in device degradation. The logarithm of n ($n_{\log}$) is assumed to follow a normal distribution:

$$n_{\log} \sim \mathcal{N}(n_{\log}|\mu_n, \sigma_n). \tag{2}$$

Here, $\mathcal{N}(x|\mu,\sigma)$ is the probability density function (PDF) of a normal distribution,

$$\mathcal{N}(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{3}$$

2.2 Failure Probability Calculation

In general, the calculation of a failure probability of a circuit is to evaluate the following integral:

$$P_{F|\boldsymbol{\phi}} = P(F|\boldsymbol{\phi}) = \int I(\boldsymbol{x};\boldsymbol{\phi})P(\boldsymbol{x}|\boldsymbol{\phi})d\boldsymbol{x}. \tag{4}$$

Here, $\boldsymbol{\phi}$ is an L-dimensional vector whose components correspond to design parameters of the circuit, such as channel length or width. $\boldsymbol{x}$ is the D-dimensional random variable that represents process variability. The joint PDF of $\boldsymbol{x}$ is given by $p_{\boldsymbol{x}}(\boldsymbol{x})$. Without loss of generality, we assume that the random variables x_d, which are the elements of $\boldsymbol{x}$, are mutually independent because any sets of random variables can be converted into uncorrelated random variables using a transformation called "whitening." Therefore, $p_{\boldsymbol{x}}(\boldsymbol{x})$ can be represented as the product of each element:

$$p_{\boldsymbol{x}}(\boldsymbol{x}) = \prod_{d=1}^{D} p_{\boldsymbol{x}}^{d}(x_d), \tag{5}$$

where $p_{\boldsymbol{x}}^{d}(\cdot)$ is a one-dimensional PDF for x_d. $I(\boldsymbol{x};\boldsymbol{\phi})$ is an indicator function that returns "1" when a realization of process variability $\boldsymbol{x}$ causes the malfunction of the circuit whose design parameters are given by $\boldsymbol{\phi}$, and returns "0" when the circuit functions correctly.

The failure event can be defined for the circuit to exceed its performance specification, such as a noise margin or a signal propagation delay, over the pre-defined threshold θ. Hence, it is natural to define the indicator function as follows:

$$I(\boldsymbol{x};\boldsymbol{\phi}) = \begin{cases} 1 & (y(\boldsymbol{x};\boldsymbol{\phi}) \geq \theta) \\ 0 & (\text{otherwise}) \end{cases}. \tag{6}$$

Here, $y(\boldsymbol{x};\boldsymbol{\phi})$ is a function which returns the performance value of the circuit. In most cases, simulators like SPICE serve as the function y.

2.3 Subset Simulation

Subset simulation (SubSim) is first proposed to efficiently calculate extremely small failure probability [4]. Note again that the naive MC method is unsuitable for estimating a small failure probability. Hence, SubSim decomposes the extremely small failure probability into a product of intermediate failure probabilities, each of which is large enough to be estimated using the naive MC method. In other words, SubSim converts a rare event simulation into a series of non-rare event simulations.

Let us consider a sequence of nested failure events $\{F = F_K \subset F_{K-1} \subset \cdots \subset F_1\}$. F is the target failure event and F_i $(i = K-1, \cdots, 2)$ is the intermediate failure event. Let Ω_k be the failure region which corresponds to the failure event F_k:

$$\Omega_k = \{\boldsymbol{x}; \quad y(\boldsymbol{x}) > \theta_k\} \ (k = 1, 2, \cdots, K), \tag{7}$$

where θ_k is an element of an increasing sequence $\boldsymbol{\theta} = \{\theta_1, \theta_2, \cdots, \theta_K\}$, which satisfies $\theta_1 < \theta_2 < \cdots < \theta_K = \theta$. Then, the failure probability $P(F)$ can be rewritten as

$$P_F = P(F) = P(F_1) \prod_{k=1}^{K-1} P(F_{k+1}|F_k) = \prod_{k=1}^{K} P_k, \tag{8}$$

where $P_1 = P(F_1)$ and $P_k = P(F_k|F_{k-1})$ for $k = 2, 3, \cdots, K$. By appropriately selecting the intermediate failure events F_k, the conditional failure probability $P(F_k|F_{k-1})$ can be made sufficiently large, so that it can be calculated using a naive MC method.

Through an example, we here describe how SubSim works. For the sake of simplicity, variable $\boldsymbol{x}$ is assumed to be two-dimensional, $\boldsymbol{x} = (x_1, x_2)$.

We first draw N random samples from $p_{\boldsymbol{x}}(\cdot)$ to obtain $\{\boldsymbol{x}^{(1,l)}; l = 1, 2, \cdots, N\}$. The performance values $\{y(\boldsymbol{x}^{(1,l)}); l = 1, 2, \cdots, N\}$ are calculated using a circuit simulator such as SPICE. Then, the T-th largest value in $\{y(\boldsymbol{x}^{(1,l)}); l = 1, 2, \cdots, N\}$ is selected and set to be θ_1. We then identify a subset of samples that satisfies $y(\boldsymbol{x}^{(1,l)}) > \theta_1$ and relabel them as $\{\boldsymbol{x}_F^{(1,t)}; t = 1, 2, \cdots, T\}$. This is an example of how the intermediate failure event F_1 is selected. Obviously, P_1 can be calculated as $P_1 = T/N$, since θ_1 is set to the T-th largest value and the size of the subset is T.

Our next objective is to calculate the conditional failure probability $P(F_2|F_1)$. To generate random samples from the conditional distribution, Markov chain Monte Carlo (MCMC) is used. A typical MCMC is based on a random walk, i.e., it forms a chain of random samples, which starts from a given seed sample. A proposal distribution and the rejection method are used to construct the chain of random samples. Samples drawn from the proposal distribution decide the move of the candidate. The rejection method then decides whether to accept the move or not, using another random sample. For the simulation of the conditional distribution, T independent chains, each of which starts from $\{\boldsymbol{x}_F^{(1,t)}; t = 1, 2, \cdots, T\}$ are formed.

In the traditional MCMCs, selection of a proposal distribution is another challenging task particularly in high-dimensional cases. To reduce the correlation among random samples, the variance of the proposal distribution should be large, which in turn leads to a low acceptance rate of the candidate walk. To solve this problem, the modified Metropolis (MM) [16] is applied.

Let us describe how MM algorithm generates the conditional samples. As mentioned above, the Markov chains start from $\boldsymbol{x}_F^{(1,t)}$. Therefore, the heads of chains, $\boldsymbol{x}^{(2,t,1)}$, are initialized as $\boldsymbol{x}_F^{(1,t)}$ for $t = 1, 2, \cdots, T$. To obtain the second sample $\boldsymbol{x}^{(2,1,2)} = [x_1^{(2,1,2)}, x_2^{(2,1,2)}]$, we first draw a random sample x_1^{new} as

$$x_1^{\text{new}} \sim q_x(x_1^{\text{new}}|x_1^{(2,1,1)}). \tag{9}$$

$q_x(\cdot|\cdot)$ is a one-dimensional proposal distribution that satisfies $q_x(x^{\text{new}}|x^{\text{old}}) = q_x(x^{\text{old}}|x^{\text{new}})$. Normal distribution is usually selected as the proposal distribution: $q_x(x_1^{\text{new}}|x_1^{(2,1,1)}) = \mathcal{N}(x_1^{\text{new}}|x_1^{(2,1,1)}, \sigma_x^2)$. The choice of σ affects the performance of MM algorithm. Practically, σ is set to the same value as the standard deviation of the manufacturing variability. We then calculate

$$r = p_{\boldsymbol{x}}^1(x_1^{\text{new}})/p_{\boldsymbol{x}}^1(x_1^{(2,1,1)}), \tag{10}$$

where $p_{\boldsymbol{x}}^1(\cdot)$ denotes the original PDF of x_1 for the manufacturing variability. A random sample u $(0 \leq u \leq 1)$ is drawn from the uniform distribution and the candidate sample $\widehat{x}_1^{(2,1,2)}$ is replaced with the new sample or with the same sample as the previous one:

$$\widehat{x}_1^{(2,1,2)} = \begin{cases} x_1^{\text{new}} & (u \leq \min(1, r)) \\ x_1^{(2,1,1)} & (u > \min(1, r)) \end{cases}. \tag{11}$$

A two-dimensional candidate sample $\widehat{\boldsymbol{x}}^{(2,1,2)}$ is obtained by calculating the other component, $\widehat{x}_2^{(2,1,2)}$ in a similar way. Then, a circuit simulation for the candidate sample is performed to calculate the corresponding performance value $\widehat{y}^{(2,1,2)} = y(\widehat{\boldsymbol{x}}^{(2,1,2)})$. Finally, $\boldsymbol{x}^{(2,1,2)}$ is set to

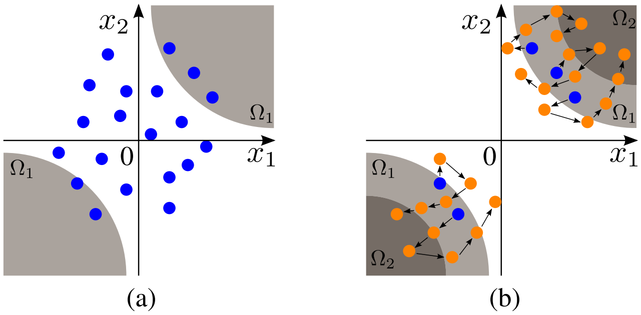


Fig. 1 A two-dimensional example to illustrate SubSim operation. (a) Naive Monte Carlo simulation to calculate P_1 in the first step of SubSim. The intermediate failure event Ω_1 is adaptively chosen. (b) Starting from the samples in Ω_1, chains of random samples are generated in the second step, and next intermediate failure event Ω_2 is again selected.

$$\boldsymbol{x}^{(2,1,2)} = \begin{cases} \widehat{\boldsymbol{x}}^{(2,1,2)} & (\widehat{y}^{(2,1,2)} \geq \theta_1) \\ \boldsymbol{x}^{(2,1,1)} & \text{(otherwise)} \end{cases}. \tag{12}$$

By repeating the above procedure, a set of conditional random samples $\{\boldsymbol{x}^{(2,1,l)}; l = 1, 2, \cdots, L\}$ are generated. Here, L denotes the length of the Markov chain and $L = N/T$. The other Markov chains starting from $\{\boldsymbol{x}^{(2,t,1)}; t = 2, 3, \cdots, T\}$ are generated and simulated in the same way. Finally, we obtain $T \cdot L$ random samples because there are T chains of length L. These procedures are illustrated in Fig. 1.

According to [4], if the Markov chain starts from an initial sample which follows the conditional distribution $P(\boldsymbol{x}|\boldsymbol{x} \in \Omega_1)$, then the generated samples also follow the same distribution. In the above procedure, the initial samples can be considered to follow the conditional distribution $P(\boldsymbol{x}|\boldsymbol{x} \in \Omega_1)$ since the initial samples are selected from random samples with a distribution $p_{\boldsymbol{x}}(\cdot)$ and that are in the first level failure region Ω_1. Therefore, the newly generated samples also follow $P(\boldsymbol{x}|\boldsymbol{x} \in \Omega_1)$. We again identify T-th largest performance value in $\{y(\boldsymbol{x}^{(2,t,l)}); t = 1, 2, \cdots, T; l = 1, 2, \cdots, L\}$ to set θ_2. The conditional failure probability is again calculated as $P_2 = P(F_2|F_1) = T/N$. The samples that belong to the next failure region $\{\Omega_2 | y(\boldsymbol{x}) \geq \theta_2\}$ are identified and MCMC is again conducted to fill the rest of $N-T$ samples distributed in Ω_2.

Repeating the above procedure until the threshold θ_k exceeds the target threshold value θ, we finally obtain the failure probability as

$$P_F = \prod_{k=1}^{K} P_k = \left(\frac{T}{N}\right)^{K-1} \cdot \frac{T'}{N}. \tag{13}$$

Here, K is the number of iterations at which θ_K exceeds θ and T' is the number of random samples that satisfy $y(\boldsymbol{x}^{(K,t,l)}) > \theta$ in the last SubSim iteration.

The selection of T is an integral part of SubSim to achieve a good sampling efficiency. According to [4], it is suggested to use 0.1, so in this paper, N and T are chosen so that the probabilities of the intermediate failure events are maintained about 0.1.

2.4 Augmented Reliability Problem

In order to calculate the failure probability given the design parameters, we first introduce an augmented reliability problem where the design parameters $\boldsymbol{\phi}$ are deliberately considered as stochastic variables whose PDF is given by $p_{\boldsymbol{\phi}}(\cdot)$. In the rest of this manuscript, we call $p_{\boldsymbol{\phi}}(\cdot)$ as "prior distribution." The role of $p_{\boldsymbol{\phi}}(\cdot)$ here is not to introduce uncertainty but to facilitate stochastic handling of $\boldsymbol{\phi}$ in the failure probability calculation framework.

In the augmented reliability problem, the failure probability $P(F|\boldsymbol{\phi})$ is considered as a conditional probability distribution given by a specific design variable $\boldsymbol{\phi}$. By applying Bayes' theorem, $P(F|\boldsymbol{\phi})$ can be rewritten as follows:

$$P(F|\boldsymbol{\phi}) = \frac{P(\boldsymbol{\phi}|F)}{P(\boldsymbol{\phi})} P(F). \tag{14}$$

Studying (14), we notice that $P(F|\boldsymbol{\phi})$ can be calculated when we know $P(F)$ and conditional probability $P(\boldsymbol{\phi}|F)$.

The calculation of $P(F)$ is the same as that described in Section 2.3 except that the pair of the process variability and the design variables $[\boldsymbol{x}, \boldsymbol{\phi}]$ are now considered to be the uncertain variables. The remaining problem is the estimation of the conditional probability $P(\boldsymbol{\phi}|F)$. Let us take a look at the failure samples which are obtained during the SubSim run. Then, we notice the failure samples distribute according to $P(\boldsymbol{x}, \boldsymbol{\phi}|F)$ and their $\boldsymbol{\phi}$ components distribute according to $P(\boldsymbol{\phi}|F)$, which means the samples that are marginalized over $\boldsymbol{x}$ component, can be used as random samples drawn from $P(\boldsymbol{\phi}|F)$:

$$P(\boldsymbol{\phi}|F) = \int P(\boldsymbol{x}, \boldsymbol{\phi}|F) d\boldsymbol{x}. \tag{15}$$

According to this observation, $P(\boldsymbol{\phi}|F)$ can be calculated by non-parametric density estimators, such as the kernel density estimator [17]. Hence, both $P(F)$ and $P(\boldsymbol{\phi}|F)$ can be obtained with a single SubSim run and using Eq. (14) yields $P(F|\boldsymbol{\phi})$.

As noted earlier, the calculation of $P(F|\boldsymbol{\phi})$ involves the estimation of the PDF of $P(\boldsymbol{\phi}|F)$ from the conditional samples, which is known to be non-parametric density estimation. In this paper, the PDF is estimated using a histogram so that Eq. (14) simply becomes:

$$P(F|\boldsymbol{\phi} \in \boldsymbol{I}_m) = \frac{P(\boldsymbol{\phi} \in \boldsymbol{I}_m|F)}{P(\boldsymbol{\phi} \in \boldsymbol{I}_m)} P(F). \tag{16}$$

Here, $P(\boldsymbol{\phi} \in \boldsymbol{I}_m)$ indicates the interval probability on a hyper-cube $\boldsymbol{I}_m$ $(m = 1, 2, \cdots, M)$ and M represents the number of bins in the histogram.

3. Proposed Method

In this section, we apply SubSim-AR to estimate the temporal change of the circuit failure probability due to device

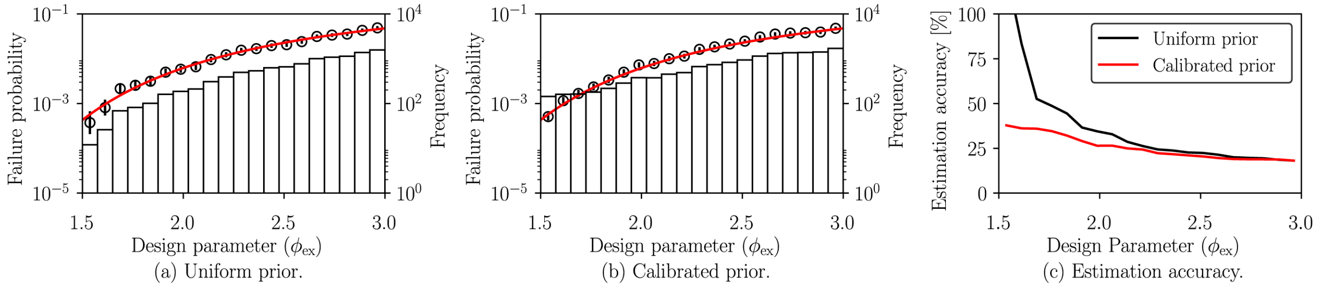


Fig. 2 Failure probability of a 1-D example calculated with SubSim-AR (markers) and the analytical method (line). The bar graph shows the histogram of ϕ_{ex} component of the failed samples.

aging. The key idea of the proposed method is to treat the device age as the design parameter ϕ. With the proposed extension, failure probability as a function of device age can be estimated through a single SubSim run. This improves computational efficiency against the existing method, in which SubSim has to be repeated many times while gradually altering the device age.

3.1 Failure Probability Calculation by Example

In this subsection, the proposed failure probability calculation is explained using a simple example.

We consider random variable r which follows a standard normal distribution whose PDF is given by (3), where $\mu = 0$ and $\sigma = 1$, and let us calculate the probability that $\phi_{\text{ex}} \cdot r$ exceeds a pre-defined threshold level C_{ex}. Here, ϕ_{ex} is a one-dimensional design parameter. In other words, the failure event is defined as $F_{\text{ex}} = \{(r, \phi_{\text{ex}}); \phi_{\text{ex}} \cdot r > C_{\text{ex}}\}$. Because the random variable r is a scalar, we can analytically calculate the conditional failure probability:

$$P\left(F_{\text{ex}}|\phi_{\text{ex}}\right) = \int_{C_{\text{ex}}/\phi_{\text{ex}}}^{\infty} p_r(r)dr = 1 - \Phi_{\text{ex}}\left(\frac{C_{\text{ex}}}{\phi_{\text{ex}}}\right), \quad (17)$$

where $\Phi_{\text{ex}}(x)$ is the cumulative distribution function (CDF) of the standard normal distribution:

$$\Phi_{\text{ex}}(x) = \int_{-\infty}^{C_{\text{ex}}/\phi_{\text{ex}}} 1/\sqrt{2\pi} \exp\left(-x^2/2\right) dx. \quad (18)$$

In this example, we approximate $\Phi_{\text{ex}}(x)$ by using the following polynomial expansion:

$$\Phi_{\text{ex}}(x) \approx 0.5 + \frac{e^{(-x^2/2)}}{\sqrt{2\pi}} \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!!}, \quad (19)$$

where !! denotes the double factorial.

Now, let us calculate $P(F_{\text{ex}}|\phi_{\text{ex}})$ using SubSim-AR to compare with that of the analytical method based on Eq. (17). Figure 2(a) shows the comparison of the two methods. Here, the target threshold C_{ex} is set to 5. Note again that we have to choose the PDF of the design parameter ϕ_{ex} before applying SubSim-AR. In this particular example, the uniform distribution on the interval [1.5, 3.0] is used. The bar graph shows the histogram of ϕ_{ex} component of the failed samples, which corresponds to $P(\boldsymbol{\phi} \in \boldsymbol{I}_m|F)$ in Eq. (16). Multiplying the augmented failure probability $P(F)$ by the conditional failure probability $P(\boldsymbol{\phi} \in \boldsymbol{I}_m|F)$, we obtain the failure probability as a function of the design parameter, which is shown as the marker with error bar indicating 95% confidence interval. The red line shows the failure probability that is analytically calculated using Eq. (17). Since the confidence interval only measures an absolute error, it is not suitable for measuring an estimation accuracy of a rare failure probability. Hence, according to [5], we utilize a normalized confidence interval as a quantitative measure of the yield estimation. Here, the normalized confidence interval is defined as the confidence interval divided by the estimated probability. Fig. 2(c) shows the estimation error using a normalized confidence interval. We notice that the histogram well approximates the change of the failure probability as a function of the design parameter and that SubSim-AR gives very good estimation of the failure probability.

From this example, we recognize the advantage of SubSim-AR over the conventional method that separately handles manufacturing variability and the device aging. In the proposed analysis, the augmented failure probability $P(F)$ is a common factor, i.e., it remains constant for different sets of design variables ϕ. In SubSim-AR, the random samples are allowed to explore the "augmented" uncertain space, consisting of the pair of the process variability and the design variables. This scheme enables all random samples contribute to the estimation of $P(F)$, increasing the calculation efficiency.

3.2 Adjustment of Prior Distribution for Design Parameter

The selection of statistical distribution for design parameter $P(\boldsymbol{\phi})$, is also an integral part to achieve higher estimation accuracy. Studying Fig. 2(a), the confidence intervals become wider as the design parameter ϕ_{ex} becomes small, which is attributed to the uniform prior given to the design parameter ϕ_{ex}. Again, SubSim-AR estimates $P(F|\boldsymbol{\phi} \in \boldsymbol{I}_m)$ by making a histogram of the ϕ component of the failed samples. If the number of failure samples in an interval $\boldsymbol{I}_m$ is small, the estimation accuracy will be drastically deteriorated. In order to overcome this issue, we propose to adjust the prior distribution so that the failure samples distribute evenly in the augmented variability space.

Before describing the detailed procedure of prior distribution adjustment, let us first demonstrate the impact of prior distribution on the estimation accuracy using the above example. Figure 2(b) shows the estimations using the calibrated prior distribution. In this example, only the prior distribution for $\boldsymbol{\phi}$ is adjusted and the other parameters, such as the number of random samples, remain unchanged. The estimation accuracy in Fig. 2(c) shows drastic improvement by the prior distribution adjustment.

Obviously, the optimal choice for the prior distribution is $P_{\mathrm{opt}}(\boldsymbol{\phi}) \propto 1/P(F|\boldsymbol{\phi})$. However, sampling from $P_{\mathrm{opt}}(\boldsymbol{\phi})$ is infeasible because it requires calculation of the conditional failure probability, $P(F|\boldsymbol{\phi})$. To resolve this chicken-and-egg problem, we propose a two-stage algorithm: in the first stage, the temporal change of failure probability is estimated coarsely to calibrate the prior distribution, and in the second stage, failure probability is accurately estimated using the calibrated distribution. The proposed two stage algorithm is summarized as follows.

(1) First stage: In the first stage, a rough sketch of the failure probability as a function of the device age is estimated by using SubSim-AR algorithm. Note again that the first stage is required only to calibrate the prior distribution and hence only the small number of circuit simulations is conducted to avoid performance overhead. SubSim-AR starts by using a uniform prior distribution. Let $P^{1\mathrm{st}}(F|\phi \in \boldsymbol{I}_m)$ be the estimated failure probability as a function of the chip age in the first stage. Then, the cumulative distribution function (CDF) of $P_{\mathrm{opt}}(\boldsymbol{\phi})$ is approximated as:

$$\Phi_m = C \cdot \sum_{i=1}^{m} \frac{1}{P^{1\mathrm{st}}(F|\phi \in \boldsymbol{I}_i)}, \tag{20}$$

where C is a normalizing constant so that $\Phi_M = 1$ is satisfied. Since a continuous rather than discrete CDF is preferable for later use, the discrete CDF is further approximated using the following model:

$$\Phi_{\mathrm{cont}}(\phi) = \frac{e^{p\phi} - 1}{e^{p} - 1}(\phi_{\max} - \phi_{\min}) + \phi_{\min}, \tag{21}$$

where p is a fitting coefficient, which is calibrated so that Eq. (21) best fits the pointwise estimation of Eq. (20). $\phi_{\min}$ and $\phi_{\max}$ are the lower and the upper limit of the design parameter, i.e., $\phi \in [\phi_{\min}, \phi_{\max}]$.

Second stage: In the second stage, SubSim-AR is conducted to accurately estimate the failure probability. Here, ϕ is sampled according to the CDF of Eq. (21) obtained in the first step. In this sample generation, any methods of random number generation can be used, such as inverse transform sampling [18], etc. Finally, the failure probability as a function of device age is estimated using Eq. (16).

3.3 Proposed Algorithm

Detailed implementation of our proposed SubSim-AR is given in Algorithms 1 and 2. In the following, the process variability $\boldsymbol{x}$ and the design variable (circuit age) ϕ are assumed to follow $p_{\boldsymbol{x}}(\cdot)$ and $p_{\phi}(\cdot)$, respectively. The failure occurs when the performance value exceeds the threshold, θ. Note that, in the proposed algorithm, ϕ is directly sampled from the prior distribution to reduce the correlation among samples.

Algorithm 1 Proposed two-stage algorithm.

```
Obtain failure samples using SubSim-AR. Here, the uniform prior distribution is given to the design variable ($\phi$), i.e. $p_{\phi}(\cdot)$ is assumed to be uniform.
Construct CDF for the prior distribution by using Eq. (20) and Eq. (21).
Conduct SubSim-AR again to estimate failure probability as a function of device age. In this step, the $\phi$ is sampled according to CDF of Eq. (21) with model parameters calibrated.
```

3.4 Calculation of Confidence Interval

The confidence interval of the estimated failure probability can be derived as below [5]. According to the central limit theorem, P_1 approximately follows the following normal distribution:

$$P_1 \sim \mathcal{N}(P_1|\mu_1, \nu_1). \tag{22}$$

Here, $\mu_1 = T/L$ and ν_1 is the sample variance which can be approximated as

$$\nu_1 \approx 1/L \cdot \mu_1(1-\mu_1). \tag{23}$$

Deriving the estimator of P_k for $k \geq 2$ is a non-trivial task because MCMC generates correlated samples. We first introduce a new random variable $s^{(k,t)}$ defined as

$$s^{(k,t)} = \frac{1}{L} \cdot \sum_{l=1}^{L} I_{F_k}(\boldsymbol{x}^{(k,t,l)}, \phi^{(k,t,l)}). \tag{24}$$

Here, $I_{F_k}(\boldsymbol{x}, \phi)$ is an indicator function that returns "1" when $y(\boldsymbol{x}, \phi) \geq \theta_k$. Otherwise, it returns "0." It is shown that P_k approximately follows the following normal distribution [5]:

$$P_k \sim \mathcal{N}(P_k|\mu_k, \nu_k). \tag{25}$$

Here, μ_k and ν_k are given by

$$\mu_k = \frac{1}{T} \cdot \sum_{t=1}^{T} s^{(k,t)}, \; \nu_k \approx \frac{1}{T(T-1)} \cdot \sum_{t=1}^{T} (s^{(k,t)} - \mu_k). \tag{26}$$

The target failure probability P_F can be obtained as the product of P_k. To convert the product into the summation, we take a logarithm of P_k. It can be shown that $\log(P_k)$ approximately follows the following normal distribution [5]:

$$\log(P_k) \sim \mathcal{N}[\log(P_k)|\log(\mu_k), \nu_{\log,k}], \tag{27}$$

where $\nu_{\log,k}$ is given by

$$\nu_{\log,k} = \nu_k/(\mu_k)^2. \tag{28}$$

Algorithm 2 SubSim-AR algorithm.

```
Generate $N$ random samples from $p_{\boldsymbol{x}}(\cdot)$ and $p_{\phi}(\cdot)$ to obtain $\{(\boldsymbol{x}^{(1,l)}, \phi^{(1,l)}); l=1, 2, \cdots, N\}$.
Calculate corresponding performance value $\{y(\boldsymbol{x}^{(1,l)}, \phi^{(1,l)}); l = 1, 2, \cdots, N\}$.
Set $\theta_1$ to $T$-th largest performance value and identify the random samples which satisfy $y(\boldsymbol{x}^{(1,l)}, \phi^{(1,l)}) \geq \theta_1$. Label these samples as $\{(\boldsymbol{x}_F^{(1,t)}, \phi_F^{(1,t)}); t=1, 2, \cdots, T\}$.
Calculate $P_1$ as $T/N$.
Set $k=2$ and $L=N/T$.
Set $(\boldsymbol{x}^{(k,t,1)}, \phi^{(k,t,1)}) = (\boldsymbol{x}_F^{(k-1,t)}, \phi_F^{(k-1,t)}); t=1, 2, \cdots, T$.
while $\theta_{k-1} < \theta$ do
    for $t=1, 2, \cdots, T$ do
        for $l=2, 3, \cdots, L$ do
            for $d=1, 2, \cdots, D$ do
                Generate random variable $x_d^{\text{new}}$ from the proposal distribution $q_{\boldsymbol{x}}(x_d^{\text{new}} | x_d^{(k,t,l-1)})$.
                Calculate $r = p_{\boldsymbol{x}}^d(x_d^{\text{new}}) / p_{\boldsymbol{x}}^d(x_d^{(k,t,l-1)})$.
                Generate a random value $u$ which uniformly distribute on [0, 1] and set $\widehat{x}_d^{(k,t,l)}$ as
                    $\widehat{x}_d^{(k,t,l)} = \begin{cases} x_d^{\text{new}} & u \leq \min(1, r) \\ x_d^{(k,t,l-1)} & \text{otherwise} \end{cases}.$
            end for
            Generate $\widehat{\phi}^{(k,t,l)}$ from $p_{\phi}(\cdot)$.
            Calculate $\widehat{y}^{(k,t,l)} = y(\widehat{\boldsymbol{x}}^{(k,t,l)}, \widehat{\phi}^{(k,t,l)})$ using a circuit simulator.
            Set $(\boldsymbol{x}^{(k,t,l)}, \phi^{(k,t,l)})$ as
                $(\boldsymbol{x}^{(k,t,l)}, \phi^{(k,t,l)})$
                $= \begin{cases} (\widehat{\boldsymbol{x}}^{(k,t,l)}, \widehat{\phi}^{(k,t,l)}) & \widehat{y}^{(k,t,l)} \geq \theta_{k-1} \\ (\boldsymbol{x}^{(k,t,l-1)}, \phi^{(k,t,l-1)}) & \text{otherwise} \end{cases}.$
        end for
    end for
    Set $\widehat{\theta_k}$ as the $T$-th largest value among $\{y^{(k,t,l)}; t = 1, 2, \cdots, T$ and $l=1, 2, \cdots, L\}$.
    if $\widehat{\theta_k} \geq \theta$ then
        Set $K=k$ and $\theta_k = \theta$.
        Identify samples that satisfy $y(\boldsymbol{x}^{(K,t,l)}, \phi^{(K,t,l)}) \geq \theta$ and label them as $\{(\boldsymbol{x}_F^{(K,t)}, \phi_F^{(K,t)}); t=1, 2, \cdots, T'\}$.
        Using the same algorithm provided above, generate another $N' = N/T'$ to populate conditional samples for accurate estimation of the conditional probability $P(\phi \in I_m | F)$.
        Calculate $P_K$ by $P_K = T'/N$.
        Break the while loop.
    else
        Set $\theta_k$ as the $T$-th largest performance value.
        Identify samples that satisfy $y(\boldsymbol{x}^{(k,t,l)}, \phi^{(k,t,l)}) \geq \theta_k$ and label them as $\{(\boldsymbol{x}_F^{(k,t)}, \phi_F^{(k,t)}); t = 1, 2, \cdots, T\}$.
        Calculate $P_k$ by $P_k = T/N$.
        Set $k = k + 1$.
    end if
end while
Calculate $P_F = \prod_{k=1}^{K} P_k$ and the histogram of $\phi$ component of the failure samples $(\boldsymbol{x}_F^{(K,t)}, \phi_F^{(K,t)})$ to obtain $P(\phi \in I_m | F)$.
$P(F | \phi \in I_m)$ is calculated using (16).
```

Then, it is shown that $\log(P_F)$ approximately follows the following normal distribution:

$$\log(P_F) \sim \mathcal{N}[\log(P_F) | \log(\mu_F), \nu_{\log,F}], \tag{29}$$

where $\log(\mu_F)$ is given by

$$\log(\mu_F) = \sum_{k=1}^{K} \log(P_k), \tag{30}$$

and $\nu_{\log,F}$ is the variance of $\log(P_F)$ whose upper bound can be approximated as [5]

$$\nu_{\log,F}^{\text{upper}} \approx \sum_{k=1}^{K} \nu_{\log,k} + 2 \cdot \sum_{k=1}^{K-1} \sqrt{\nu_{\log,k} \cdot \nu_{\log,k+1}}. \tag{31}$$

We then derive the confidence interval of $P(\phi \in I_m | F) = P_{\phi|F}^m$ in a similar way. We first consider a new random sample:

$$s_{\text{cond}}^{(t,m)} = \frac{1}{L} \sum_{l=1}^{L} I_m(\phi^{(K,t,l)}), \tag{32}$$

where $I_m(\phi)$ returns "1" when $\phi \in I_m$ and "0" when $\phi \notin I_m$. Note that the ϕ component of the failed samples is used here. Just like the estimation of P_k, it can be shown that the logarithm of the conditional probability $\log(P_{\phi|F}^m)$ approximately follows normal distribution:

$$\log(P_{\phi|F}^m) \sim \mathcal{N}[\log(P_{\phi|F}^m) | \log(\mu_{\phi|F}^m), \nu_{\log,\phi|F}^m]. \tag{33}$$

Here, $\mu_{\phi|F}^m$ and $\nu_{\log,\phi|F}^m$ are given by

$$\mu_{\phi|F}^m = \frac{1}{T'} \sum_{t=1}^{T'} s_{\text{cond}}^{(t,m)} \tag{34}$$

and

$$\nu_{\log,\phi|F}^m \approx \frac{1}{T'(T'-1)(\mu_{\phi|F}^m)^2} \sum_{t=1}^{T'} (s_{\text{cond}}^{(t,m)} - \mu_{\phi|F}^m). \tag{35}$$

By combining the logarithm of Eq. (16), and Eqs. (29) and (33), we finally obtain the approximation of the PDF of $\log P(F | \phi \in I_m) = \log P_{F|\phi}^m$ as follows:

$$\log(P_{F|\phi}^m) \sim \mathcal{N}[\log(P_{F|\phi}^m) | \log(\mu_{F|\phi}^m), \nu_{\log,F|\phi}^m], \tag{36}$$

where $\log(\mu_{F|\phi}^m)$ and $\nu_{\log,F|\phi}^m$ are given by

$$\log(\mu_{F|\phi}^m) = \log(\mu_F) + \log(\mu_{\phi|F}^m) - \log P_{\phi}^m, \tag{37}$$

and

$$\nu_{\log,F|\phi}^m \approx \nu_{\log,F} + \nu_{\log,\phi|F}^m + 2\sqrt{\nu_{\log,F} \cdot \nu_{\log,\phi|F}^m}. \tag{38}$$

Here, $\log P_{\phi}^m$ is the logarithm of the probability that the design variable falls in the region I_m, which can be analytically calculated because ϕ is an artificial random variable having a known (given) distribution. In this implementation, we selected a uniform distribution for $p_{\phi}(\cdot)$. Therefore, $\log(P_{\phi}^m)$ can be obtained as $\log(1/M) = -\log(M)$, where M is the number of bins for the histogram. Finally, we can derive the confidence interval using the above equations. For example, the 95% confidence interval can be represented as

$$\left\{ \exp\left[\log(\mu_{F|\phi}^m) - 1.96\sqrt{\nu_{\log,F|\phi}^m} \right], \right.$$
$$\left. \exp\left[\log(\mu_{F|\phi}^m) + 1.96\sqrt{\nu_{\log,F|\phi}^m} \right] \right\}. \tag{39}$$

4. Numerical Experiment

4.1 Experimental Setup

In this section, the proposed method is validated through the numerical experiments of calculating the clock-to-q delay degradation of a DFF. The schematic of the DFF is shown in Fig. 3. The failure is defined as an event that clock-to-q delay to exceed a given threshold. In order to simulate the process variation, we assume variabilities in the initial V_{TH}, gate length, and gate width. For a pMOS transistor, we additionally consider the variability in NBTI-induced degradation. In total, 84 independent random variables are used in this experiment.

The initial V_{TH} variation of each transistor is assumed to follow a normal distribution:

$$\Delta V_{\mathrm{TH}}^{\mathrm{INIT}} \sim \mathcal{N}(0, A_{V_T}/\sqrt{L \cdot W}). \tag{40}$$

Here, A_{V_T} is the Pelgrom coefficient with an assumed value of 4×10^{-9} V/m^2. L and W are the channel length and width, respectively. The variations of gate length and width are assumed to follow normal distributions whose standard deviations are 1 nm and 5 nm, respectively.

As the aging mode, we consider the NBTI-induced V_{TH} shift ($\Delta V_{\mathrm{TH}}^{\mathrm{NBTI}}$) according to (1). Model parameters, such as k, and the mean and standard deviation of the log-normal distribution in (2), are adjusted so that the model well reflects degradations and the variability observed on a commercial 65 nm process. Note again that in the first SubSim run, a uniform distribution is given to ϕ. Then, in the second run, the distribution is modified so that the failure samples uniformly distribute over the design variable space.

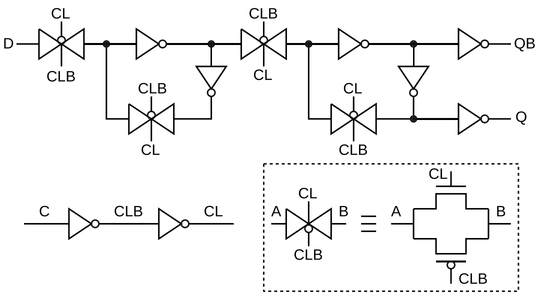


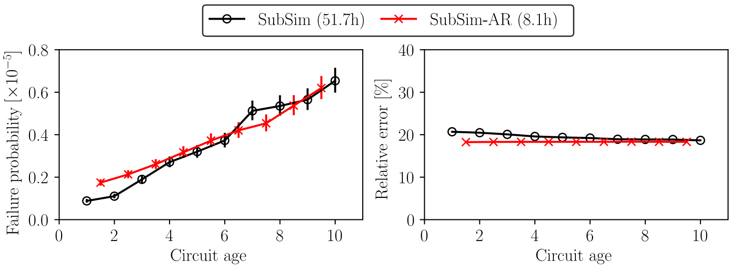
Fig. 3 Circuit schematic of a DFF.

4.2 Result

We calculated the change of the failure probability as a function of device age. The red markers in Fig. 4(a) show the results of the proposed method based on the augmented reliability approach, SubSim-AR. The x- and y-axis correspond to the device age and failure probability, respectively. The error bar shows the 95% confidence interval. The required calculation time is also shown inside the bracket of labels. In our experiment, all numerical calculations are conducted on a Linux PC of Intel Xeon E7-4870. As a comparison, the failure probability shift of the same circuit is estimated by repeating SubSim at each aging time step from 1.0 to 10.0 years at the interval of 1 year, whose result is shown using the black marker. Hereafter, we call this conventional approach which requires multiple SubSim runs as SubSim-Multi.

The proposed method calculates the failure probability in the intervals. For example, the leftmost red marker shows the averaged failure probability over the first 1 year. On the other hand, SubSim-Multi calculates the failure probability at the exact time point, i.e., the leftmost black marker shows the failure probability of the 1-year-old circuit. The selection of the time interval width is practically important since it affects both on the time resolution of the estimated yield and on the estimation accuracy: reducing the width of the time interval improves the time resolution at the cost of the reduced estimation accuracy, when the total number of the samples is unchanged. The interval width is adjusted so as to satisfy a required estimation accuracy while avoiding a large change in the estimated yield among different time steps. In order to predict whether the selected interval width satisfies the above requirements, the outcome of the first stage is again utilized. In our particular experiment, an interval of one year should be sufficient to see the degradation of the failure probability. Studying Fig. 4(a), we notice that the probabilities obtained by SubSim-Multi do not just suffer from large variance, but exhibit non-monotonic changes along the device age, which also demonstrates the advantage of SubSim-AR over SubSim-Multi.

Figure 4(b) shows the relative error defined as the ratio



(a) Calculated failure probability degradation.

(b) Comparison of the accuracy.

Fig. 4 Calculated failure probability (a) and its accuracy (b).

of the 95% confidence interval to the estimated failure probability. The red and the black markers again show the results of SubSim-AR and SubSim-Multi, respectively. Studying Fig. 4(b), the estimation accuracy of SubSim-AR is almost constant, owing to the proposed two-stage algorithm. Judging from error bars, the estimation accuracy of the proposed method is almost equal to or slightly higher than that of SubSim-Multi. Considering the computation time difference, the proposed method achieved 6.38× speed up compared to the conventional method.

5. Conclusion

In this paper, an efficient method to evaluate the impact of device aging on the failure probability is proposed. The augmented reliability approach, which was originally proposed to understand the impact of design parameters in structural design area, was extended to analyze failure probability of the circuits due to device aging. By considering the age of the circuit as a design parameter, an efficient calculation of the degradation of failure probability has become possible. In order to enhance accuracy for the ages of analysis, a two-stage algorithm was proposed to adjust the parameters for prior distribution. The simulation experiment showed that the proposed method (SubSim-AR) achieved 6.38× speed up over the conventional method with almost equal accuracy.

Acknowledgments

This work was partially supported by a Grant-in-Aid for JSPS Fellows and MEXT/JSPS KAKENHI Grant No. 26280014 and 17H01713. The authors also acknowledge support from VDEC with the collaboration with Synopsys Corporation.

References

[1] A. Bhavnagarwala, X. Tang, and J. Meindl, "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," IEEE J. Technol. Computer Aided Des., vol.36, no.4, pp.658–665, 2001.

[2] 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," Design Autom. Conf., pp.69–72, 2006.

[3] K. Katayama, S. Hagiwara, H. Tsutsui, H. Ochi, and T. Sato, "Sequential importance sampling for low-probability and high-dimensional SRAM yield analysis," Int. Conf. Comput. Aided Design, pp.703–708, 2010.

[4] S.K. Au and J.L. Beck, "Estimation of small failure probabilities in high dimensions by subset simulation," Probabilistic Eng. Mechanics, vol.16, no.4, pp.263–277, 2001.

[5] S. Sun and X. Li, "Fast statistical analysis of rare circuit failure events via subset simulation in high-dimensional variation space," Int. Conf. Comput. Aided Design, pp.324–331, 2014.

[6] E. Maricau and G. Gielen, "Efficient variability-aware NBTI and hot carrier circuit reliability analysis," IEEE J. Technol. Computer Aided Des., vol.29, no.12, pp.1884–1893, 2010.

[7] E. Maricau and G. Gielen, "Stochastic circuit reliability analysis," Design Automation and Test in Europe, pp.1–6, 2011.

[8] K. Kang, H. Kufluoglu, K. Roy, and M. Alam, "Impact of negative-bias temperature instability in nanoscale SRAM array: Modeling and analysis," IEEE J. Technol. Computer Aided Des., vol.26, no.10, pp.1770–1781, 2007.

[9] Y.L. Chen, W. Wu, C.N. Liu, and L. He, "Incremental Latin hypercube sampling for lifetime stochastic behavioral modeling of analog circuits," Asia South Pasific Design Autom. Conf., pp.556–561, 2015.

[10] S.K. Au, "Reliability-based design sensitivity by efficient simulation," Computers Structures, vol.83, no.14, pp.1048–1061, 2005.

[11] T. Grasser, W. Gos, V. Sverdlov, and B. Kaczer, "The universality of NBTI relaxation and its implications for modeling and characterization," Int. Rel. Phys. Symp., pp.268–280, April 2007.

[12] B. Kaczer, V. Arkhipov, R. Degraeve, N. Collaert, G. Groeseneken, and M. Goodwin, "Disorder-controlled-kinetics model for negative bias temperature instability and its experimental verification," Int. Rel. Phys. Symp., pp.381–387, April 2005.

[13] K. Sutaria, J. Velamala, C. Kim, T. Sato, and Y. Cao, "Aging statistics based on trapping/detrapping: Compact modeling and silicon validation," IEEE Trans. Device Mater. Rel., vol.14, no.2, pp.607–615, 2014.

[14] H. Awano, S. Morita, and T. Sato, "Scalable device array for statistical characterization of BTI-related parameters," IEEE Trans. Very Lagre Scale Integr. (VLSI) Syst., vol.25, no.4, pp.1455–1466, April 2017.

[15] H. Awano, M. Hiromoto, and T. Sato, "Variability in device degradations: Statistical observation of NBTI for 3996 transistors," European Solid State Device Research Conf., pp.218–221, 2014.

[16] S.K. Au and J.L. Beck, "Subset simulation and its application to seismic risk based on dynamic analysis," J. Eng. Mechanics, vol.129, no.8, pp.901–917, 2003.

[17] B.W. Silverman, Density Estimation for Statistics and Data Analysis, CRC press.

[18] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

Hiromitsu Awano received his B.E. degree in Informatics and M.Sc. and Ph.D. degrees in Communications and Computer Engineering from Kyoto University in 2010, 2012, and 2016, respectively. He was with Hitachi, Ltd., Tokyo, Japan in 2016. In 2017, he joined the VLSI Design and Education Center, The University of Tokyo, Japan, where he is an assistant professor. His research interests include CAD for VLSI design and hardware accelerator for machine learning. He was a research fellow of japan society for the promotion of science and a member of IPSJ.

Takashi Sato received B.E. and M.E. degrees from Waseda University, Tokyo, Japan, and a Ph.D. degree from Kyoto University, Kyoto, Japan. He was with Hitachi, Ltd., Tokyo, Japan, from 1991 to 2003, with Renesas Technology Corp., Tokyo, Japan, from 2003 to 2006, and with the Tokyo Institute of Technology, Yokohama, Japan. In 2009, he joined the Graduate School of Informatics, Kyoto University, Kyoto, Japan, where he is currently a professor. He was a visiting industrial fellow at the University of California, Berkeley, from 1998 to 1999. His research interests include CAD for nanometer-scale LSI design, fabrication-aware design methodology, and performance optimization for variation tolerance. Dr. Sato is a member of IEICE. He received the Beatrice Winner Award at ISSCC 2000.