IEICE TRANS. FUNDAMENTALS, VOL.E99–A, NO.7 JULY 2016

PAPER Special Section on Design Methodologies for System on a Chip

Efficient Aging-Aware SRAM Failure Probability Calculation via Particle Filter-Based Importance Sampling

Hiromitsu AWANO[†a)], ***Student Member***, **Masayuki HIROMOTO**[†], ***and*** **Takashi SATO**[†], ***Members***

SUMMARY An efficient Monte Carlo (MC) method for the calculation of failure probability degradation of an SRAM cell due to negative bias temperature instability (NBTI) is proposed. In the proposed method, a particle filter is utilized to incrementally track temporal performance changes in an SRAM cell. The number of simulations required to obtain stable particle distribution is greatly reduced, by reusing the final distribution of the particles in the last time step as the initial distribution. Combining with the use of a binary classifier, with which an MC sample is quickly judged whether it causes a malfunction of the cell or not, the total number of simulations to capture the temporal change of failure probability is significantly reduced. The proposed method achieves 13.4× speed-up over the state-of-the-art method.

key words: *SRAM cell yield, failure probability calculation, NBTI, importance sampling, particle filter, Monte Carlo method*

1. Introduction

Miniaturization of semiconductor devices have enabled manufactures to integrate billions of transistors into a single silicon chip. On the other hand, as the manufacturing variability continues to increase, circuit design using such "unreliable" components is increasingly becoming a difficult challenge. One example of such challenges is a bit cell design of a static random access memory (SRAM). Considering the fact that a modern microprocessor embeds tens of mega bytes of on-chip cache, extremely low failure probability is required for a single SRAM cell. A typical failure probability required for an SRAM cell is reported to be as low as 10^{-8} to 10^{-6}, or below [1].

Numerical estimations of such small failure probability is known to be a difficult task. A naive Monte Carlo (MC) method, which directly generates random samples in a variability space, requires millions or billions of circuit simulations to obtain only a single failure sample. Hence, it is almost impossible for the naive MC methods to accurately calculate the small failure probability. Importance sampling techniques are definitely required to overcome this problem [2]–[4].

The variability of transistor-parameters is mostly originated in the course of manufacturing process. As the shrinkage of semiconductor manufacturing process continues, even an atomic level bump on a gate terminal or a fluctuation of the number of dopant ions have large impact on the electrical property of transistors. In addition to such "static" variability, we are currently forced to cope with an increasing impact of "dynamic" variability that originates from transistor aging. Thin gate-oxide layer in highly scaled transistors poses various new problems on the reliability of large-scale integration (LSI) circuits. Among various degradation mechanisms, negative bias temperature instability (NBTI) is an increasing concern [5]. NBTI is observed as gradual increase in the threshold voltage (V_{TH}) of transistors. Operational temperature, supply voltage, and stress period are the three major factors determining the magnitude of NBTI induced V_{TH} shift: high operational temperature, high stress voltage, and long stress period promote the NBTI-induced V_{TH} degradation. To improve the reliability of the LSI, designers must take the impact of the NBTI induced device degradation into consideration as early in the design phase as possible. Development of computer aided design (CAD) tools that accurately evaluate and countermeasure the device degradation has thus emerged as an urgent issue.

SRAM cells are considered to be one of the most vulnerable circuit components to the NBTI-induced V_{TH} shift for two reasons. First, a long stress period is frequently observed. Because of data-storage functionality, switching activity of an SRAM cell is usually lower than that of combinational circuits. Hence, a pMOS transistor in one of the coupled inverters is more likely to be exposed to constant stress, deteriorating the stability of the cell. Second, the heat produced by components surrounding an SRAM cell further complicates the problem. Although the SRAM cell itself produces only small amount of heat, components such as register files are typically surrounded by highly active circuits, such as instruction dispatchers, reorder buffers, etc. Heat generated by these components accelerates the NBTI-induced degradation. Circuit designers have to be extremely careful to optimize the circuit structure of an SRAM cell and memory placement in order to improve the total reliability of LSI.

In this paper, we propose a novel and efficient failure probability calculation method of SRAM cells under the NBTI stress. A considerable amount of efforts have been paid to accelerate the failure probability calculation of an SRAM cell [2]–[4]. Those methods, however, only consider the static variability such as the one caused in the manufacturing process and invariant thereafter. In order to keep track the change of failure probability due to dynamic variability, failure probability calculations have to be conducted

Manuscript received September 17, 2015.
Manuscript revised January 19, 2016.

[†]The authors are with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606-8501 Japan.
a) E-mail: awano@easter.kuee.kyoto-u.ac.jp

DOI: 10.1587/transfun.E99.A.1390

repetitively at different aging time. Even if a single failure probability calculation is accelerated using advanced sampling techniques, a large simulation effort is still required.

Our proposed method solves this problem by applying the concept of the particle filter to incrementally track the time-changing characteristics of an SRAM cell. The concept of particle filter is first introduced into the CAD community in [6] to enhance the efficiency of the importance sampling (IS) based failure probability calculation. In the IS-based MC, random samples are generated from the distorted distribution, which is called "alternative distribution," not from the original distribution. The alternative distribution is selected such that more failure samples are drawn (i.e., samples drawn from the alternative distribution is more likely to cause circuit failure). Due to the complicated shape of the pass/fail border, its analytical representation is difficult to obtain while the approximation of the distribution using a simple distribution will deteriorate the efficiency. Hence, in [6], particles were used to represent the complex shape of the alternative distribution to achieve drastic speed-up over conventional importance sampling approaches. Our contribution in this paper is to extend the method in [6] so as to efficiently handle the aging effect. Due to NBTI-induced device degradation, the shape of the optimal alternative distribution changes as device ages. Because construction of the alternative distribution from scratch is a computationally heavy task, we exploit the characteristics of the NBTI-induced device degradation. Specifically, when the change of V_{TH} is gradual, the change of the optimal alternative distribution is also gradual. Hence, in the proposed method, the temporal change of the optimal alternative distribution is followed by the particles moving around the variability space. This procedure substantially accelerates the total calculation time of failure probability along aging time steps; it eliminates the independent explorations of the optimal alternative distribution for different aging time.

We also integrate a support vector machine (SVM) based binary classifier with the particle filter. Firstly, the binary classifier is trained using a small subset of random samples to roughly judge whether a sample causes circuit failure or not. Using the classifier, the majority of the random samples are classified as either pass or fail without executing time-consuming transistor-level simulations. Because the classifier is based on a linear model, the time required for the classifications is negligibly small. The reduction of the total calculation time is significant even though the time to train the classifier is newly introduced.

An adoption of a two-stage MC flow further reduces the calculation time while maintaining accuracy. In the first stage, a rough estimation of the optimal alternative distribution is obtained using a small number of random samples. Then, in the second stage, a failure probability is accurately calculated using the samples generated from the alternative distribution. With the above modifications, our method achieves 1.8× speed-up of the failure probability calculation on a single aging time step compared to the state-of-the-art method [6]. Total calculation time required to obtain the temporal change of the failure probability is further reduced, achieving 13.4× speed up compared to the conventional method.

The rest of this paper is constructed as follows. In Sect. 2, related works are reviewed. In Sect. 3, we explain background that forms the basis of our method. In Sect. 4, we explain the details of the proposed method. Then in Sect. 5, we describe experimental procedures and its results. Finally in Sect. 6, conclusion remarks are provided.

2. Related Work

There are frameworks that analyze the impact of NBTI on the circuit operation. In [7] and [8], the methods that can consider both static and dynamic variability is proposed. In [9], the impact of NBTI on the stability of an SRAM cell is analyzed and an efficient method to estimate the failure probability of the cell is proposed. However, those methods are based on approximation models. In [7] and [8], a response surface model (RSM) is used to approximate the circuit response to process variability. In [9], a noise margin of an SRAM cell is assumed to follow a normal distribution and the failure probability is calculated under that assumption. A normal distribution provides a good approximation of the target distribution around its average. On the other hand, its rightmost tail is not so accurate. As we mentioned, the failure probability required for modern SRAM cells is extremely low and hence rightmost tail of the distribution must be analyzed very accurately. Introduction of such approximations may lead to the probability estimation that is unacceptably inaccurate.

MC-based methods are therefore required to accurately analyze the failure probability. The naive MC is one of the most popular way to estimate the probability, in which random samples corresponds to variabilities of transistors are drawn from a probabilistic distribution and transistor level simulations are performed to see whether those variabilities cause malfunctions of the circuit or not. Theoretically, the naive MC can give an accurate estimation of the failure probability. However, in solving the problem with extremely small failure probability, millions or billions of circuit simulations are required to obtain sufficient number of failed samples and hence the naive MC can not calculate the failure probability in a reasonable time. To solve this problem, importance sampling techniques are usually used [2]–[4]. The selection of the alternative distribution in importance sampling is crucial to the acceleration rate of the failure probability calculation. Because determination of a good alternative distribution requires significant computational efforts, many attempts are made to accelerate the construction process. Authors of [10] proposed a mean-shift method in which the alternative distribution is approximated with a distribution whose mean is shifted to the most probable failure point. In [11], a method known as "Markov chain Monte Carlo (MCMC)" is used to explore the process variability space efficiently. Authors of [6] utilized particles that moves in the process variability space to automatically

construct an alternative distribution.

In order to see how the failure probability changes over time, multiple failure probability calculations are required. Because conventional methods [6], [10], [11] do not consider the effect of NBTI-induced device degradation, the alternative distribution is constructed from scratch at each repetitive calculation. Here, we notice that the NBTI-induced device degradation is a gradual process and hence the reconstruction of the alternative distribution is clearly inefficient. Therefore, we propose to "reuse" the particles that lie close to pass/failure boundary in the calculation of the most recent aging time step. This enables the particles incrementally track the temporal change of the alternative distribution. Multiple explorations in the process variability space are thus eliminated, contributing the increase of the efficiency.

3. Background

3.1 Failure Probability Calculation

Failure probability calculation is generally formulated as

$$P_{\text{fail}} = \int I(\boldsymbol{x})P(\boldsymbol{x})d\boldsymbol{x}. \tag{1}$$

Here, P_{fail} is the failure probability and $\boldsymbol{x}$ is the D-dimensional random variable which corresponds to random variations of the transistor parameters, such as V_{TH}, channel length, gate oxide thickness, etc. $P(\boldsymbol{x})$ is the probability density function (PDF) over the process variability. $I(\boldsymbol{x})$ is an indicator function that returns "1" if the given random variable $\boldsymbol{x}$ causes a malfunction of an SRAM cell, and "0" otherwise. We hereafter call regions in which failure samples ($\boldsymbol{x}_{\text{fail}}$ such that $I(\boldsymbol{x}_{\text{fail}}) = 1$) distribute as "failure regions."

Because the indicator function does not have an analytical form in general, we rely on an MC approximation to evaluate (1). The above integral is calculated using random samples drawn from $P(\boldsymbol{x})$, i.e. $\boldsymbol{x}_i \sim P(\boldsymbol{x})$:

$$P_{\text{fail}} \approx \frac{1}{N}\sum_{i=1}^{N} I(\boldsymbol{x}_i). \tag{2}$$

A naive MC method in (2) can not be applied to the calculation of (1) in low failure probability problems because very few or no samples that cause a malfunction of an SRAM cell can be generated in practical time.

In order to improve the sampling efficiency, importance sampling techniques is developed. The key idea of the importance sampling is to calculate (1) using samples drawn from an alternative distribution $Q(\boldsymbol{x})$. The following equation is obtained by modifying (1):

$$P_{\text{fail}} = \int I(\boldsymbol{x})\frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})}Q(\boldsymbol{x})d\boldsymbol{x}. \tag{3}$$

The MC approximation of (3) using the samples drawn from $Q(\boldsymbol{x})$ is obtained as:

$$P_{\text{fail}} \approx \frac{1}{N}\sum_{i=1}^{N} I(\boldsymbol{x}_i)\frac{P(\boldsymbol{x}_i)}{Q(\boldsymbol{x}_i)}. \tag{4}$$

The optimal alternative distribution is known to be

$$Q_{\text{opt}}(\boldsymbol{x}) \propto I(\boldsymbol{x})P(\boldsymbol{x}). \tag{5}$$

If we can draw samples from $Q_{\text{opt}}(\boldsymbol{x})$, a perfect approximation of (3) with zero variance can be achieved because $I(\boldsymbol{x})P(\boldsymbol{x})/Q_{\text{opt}}(\boldsymbol{x})$ becomes a constant. This means that, in order to improve the efficiency, we have to select $Q(\boldsymbol{x})$ whose shape is close to $Q_{\text{opt}}(\boldsymbol{x})$. However, this is an infeasible task because we do not know the exact shape of the indicator function $I(\boldsymbol{x})$. We here introduce a particle filter to enable an automatic estimation of the optimal alternative distribution.

3.2 Particle Filter

Particle filter is an on-line estimator of non-Gaussian distributions [12], [13]. A probabilistic density is approximated using the density of particles that move in the D-dimensional variability space. The positions of the particles are updated iteratively using the following steps as shown in Fig. 1.

Prediction

The locations of the candidate particles in the next iteration are drawn from the proposal distribution $q(\boldsymbol{x})$. A usual choice is a mixture of normal distributions with each component centered at each position of the particles generated in the previous iteration so that the regions where the previous particles existed are more likely to be visited.

Measurement

The weights that represent the goodness of fit of each candidate particle are calculated. In the context of the failure probability calculation of an SRAM cell, the weight is calculated as $I(\boldsymbol{x}) \cdot P(\boldsymbol{x})$. In case that $P(\cdot)$ is a normal distribution, large weights are assigned to the particles that are in the failure region and close to the origin of the variability space.

Resampling

The particles are resampled from the candidate particles ac-

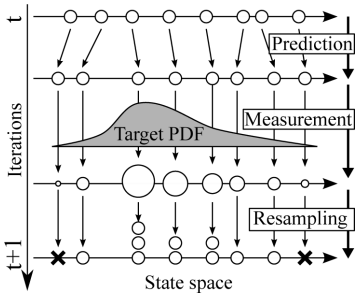


Fig. 1 Particle filter.

cording to the probabilities proportional to the weight assigned in the **Measurement step**. Hence, the candidate particles with the larger weights attain more number of copies while the particles with the smaller weights attain smaller number of copies. Those outside the failure region are eliminated because $I(\boldsymbol{x})$ returns "0" for those samples and their weights become zero. The particle density becomes gradually closer to the distribution $I(\boldsymbol{x}) \cdot P(\boldsymbol{x})$ by repeating the above procedures. The approximation of the optimal alternative distribution can be obtained as the distribution of particles.

3.3 Support Vector Machine

Although a large portion of the samples generated from the particle-approximated alternative distribution are in the failure region, a small amount of pass samples are also generated due to the approximation error. If we could obtain the optimal alternative distribution, all of the random samples drawn from the distribution will be the failure samples and hence no transistor-level simulations to calculate $I(\boldsymbol{x})$ are required. Unfortunately, however, only an approximation of the optimal alternative distribution can be obtained in practice and the evaluations of the indicator function are required for all of the random samples generated.

Each time the indicator function $I(\boldsymbol{x})$ is evaluated, a transistor-level simulation is performed. This step occupies almost all of the total calculation time because the transistor-level simulation is a computationally heavy task. To accelerate the calculation of $I(\boldsymbol{x})$, we introduce a binary classifier based on a support vector machine (SVM). SVM is a supervised training model for binary classification [14]. Given a set of training examples that consist of feature vectors and corresponding labels, SVM learns a classification model which categorizes a new feature vector into one of the two classes, i.e., pass or fail.

SVM assumes a linear classification model:

$$c = \sum_i w_i f_i. \tag{6}$$

Here, w_i is a coefficient of a particular feature quantity and f_i is the i-th element of a feature vector $\boldsymbol{f}$. The signature of c represents the class label of the feature vector. In other words, SVM learns a hyper plane in a feature space, which separates training examples into two classes as shown in Fig. 2. A good separation for a new feature vector is achieved when the distances between training examples and the hyper-plane are the largest.

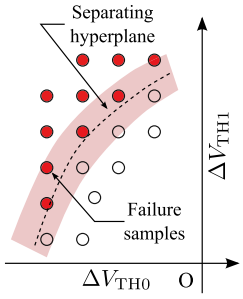


Fig. 2 Support vector machine.

3.4 NBTI-Induced V_{TH} Shift and Its Variability

NBTI is a gradual V_{TH} increase observed on pMOS transistors. When a negative bias is applied to the gate terminal of the pMOS transistor, its V_{TH} starts to increase gradually. Most part of the increased V_{TH} recovers as soon as the transistor is released from the stress state. However, there is an unrecoverable component of degraded V_{TH}. Hence, repeatedly applying stress will gradually increase V_{TH} of the pMOS transistor and will finally cause a malfunction of the circuit.

In spite of the intensive researches on NBTI, its physical mechanism is still a controversial topic. Currently proposed physical models of NBTI are divided into two groups: one based on a reaction diffusion (RD) theory and the other based on a trap detrap (TD) theory. RD-theory explains the NBTI induced V_{TH} shift as follows. First, a negative bias applied to the gate terminal forms an electric field, which breaks a Si-H bond at the silicon-oxide interface. Then, the released H atom migrates in the oxide and forms a fixed positive charge, which contributes to the V_{TH} shift. The remaining bond of a silicon (Si-) captures an electron, which also contributes to the V_{TH} shift. Based on this model, the relationship between device age and V_{TH} shift can be written using a power law model [15]:

$$\Delta V_{TH}^{NBTI} = k \cdot t_{age}^{n}. \tag{7}$$

Here, k is a model parameter which reflects the stress condition, operational temperature, and fabrication process. t_{age} is an age of a transistor. n is also a model parameter that varies from transistor to transistor [15].

Meanwhile, in small transistors, researchers noticed a stair-like recovery of V_{TH} when a transistor is released from the negative stress condition [16]. This observation leads to a TD-theory, in which pre-existing defects located inside the gate oxide film are considered to be the origin of the V_{TH} shift [17]. When a negative bias is applied, the defects capture electrons, causing V_{TH} to increase. The defects then release the electrons when the transistor is released from the stress condition and the degraded V_{TH} starts to recover. The following is a compact model derived from the TD-theory [15]:

$$\Delta V_{TH}^{NBTI} = \phi \left[A + \log(1 + C \cdot t_{age}) \right]. \tag{8}$$

Here, A and C are parameters that reflect the stress condition or a manufacturing process, and thus they are relatively constant for the transistors on a same chip. ϕ is also a model parameter reflecting the number of defects included in the transistor.

Now, let us take a look at actual V_{TH} shifts acquired in a silicon measurement. Figure 3 shows examples of NBTI-

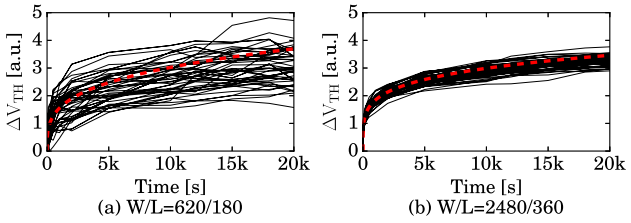


Fig. 3 Examples of NBTI-induced V_{TH} shift observed in 50 pMOS transistors.

induced V_{TH} shifts observed from 50 pMOS transistors fabricated using a commercial 180-nm CMOS process [18]. The results of two sizes of transistors are shown. The V_{TH} shift varies widely among transistors just like the initial V_{TH} variation. It is clear from Figs. 3(a) and (b) that V_{TH} shift of smaller transistors varies widely while the overall trend of the degradations on both types of transistors are almost the same. The objective of this study is to propose a failure probability calculation method which can handle NBTI-induced device degradation. However, as it is clear from Fig. 3(a), we are also required to take the variability in device degradation into account.

In the compact model equations in (7) and (8), the model parameters that reflect the variability of the degradation are n for the RD-based model and ϕ for the TD-based model [15]. In this study, we employ the RD-based model because it is simpler than the TD-based model and easy to extract the model parameters from silicon measurements. However, since our proposal is based on an MC approach, other models such as the TD-based model can be used completely in the same way.

The red lines in Fig. 3 show the averaged model prediction over the 50 transistors. We can see that the model well predicts the temporal change of V_{TH}. This justifies the use of the RD-model for long-term reliability assessment. The remaining problem is to find the statistical distribution, to which the parameter n follows. It is experimentally shown that n follows a log-normal distribution [16], [18]. According to the measurement result, we employ a log-normal distribution for the statistical distribution of n. Hence, the logarithm of n ($n_{\log}$) is assumed to follow a normal distribution:

$$n_{\log} \sim \mathcal{N}(n_{\log}|\mu_n, \sigma_n). \tag{9}$$

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

$$\mathcal{N}(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{10}$$

4. Proposed Method

4.1 Variability and Degradation Modeling

Let us first describe the variability and degradation modeling. We here deal with variabilities in V_{TH} but other sources of variabilities, such as those in a channel size or in a gate oxide thickness, can be easily taken into account. In [19], the authors measured 11 billion transistors and showed that the V_{TH} variation of fresh transistors follows a normal distribution. Based on this evidence, we employ a normal distribution for the variability model of V_{TH}. Hence, V_{TH} variation of a fresh transistor is given by

$$\Delta V_{\mathrm{TH}}^{\mathrm{fresh}} = \frac{A_{V_{\mathrm{TH}}}}{\sqrt{L \cdot W}} x^{\mathrm{fresh}}. \tag{11}$$

Here, $A_{V_{\mathrm{TH}}}$ is a Pelgrom coefficient, and L and W are channel length and width, respectively. x^{fresh} is a random variable which is assumed to follow the standard normal distribution:

$$x^{\mathrm{fresh}} \sim \mathcal{N}\left(x^{\mathrm{fresh}} \middle| 0, 1\right). \tag{12}$$

The NBTI-induced V_{TH} shift is given by (7). Note again that n is a random variable that represents degradation variation. As stated in Sect. 3.4, n is represented using a log-normal distribution. Hence, a logarithm of n ($n_{\log}$) is given by

$$n_{\log} = x^{\mathrm{bti}} \cdot \sigma_n + \mu_n. \tag{13}$$

Here, x^{bti} is again a random variable following the standard normal distribution. σ_n and μ_n are the standard deviation and mean of the distribution of $n_{\log}$.

A single MC trial proceeds as follows. A set of random samples are drawn from a probability distribution. In the failure probability calculation of an SRAM cell, there are eight random variables: six for the V_{TH} variation of six transistors and other two are NBTI-induced degradation of the two pMOS transistors. We then calculate the corresponding V_{TH} shift of each transistor using (11), (7), and (13). Note that simple sum of $\Delta V_{\mathrm{TH}}^{\mathrm{fresh}}$ and $\Delta V_{\mathrm{TH}}^{\mathrm{NBTI}}$ gives a total V_{TH} shift for a pMOS transistor, because the V_{TH} variation of fresh transistors and their degradations are reported to be independent [18]. Then, performance of the circuit (e.g. noise margin) with the variability of the transistors is calculated using a transistor-level simulator such as SPICE. Finally, a pass or fail label, i.e., the value of the index function $I(\boldsymbol{x})$, is obtained using the calculated performance value.

4.2 Overview of the Proposed Method

A failure probability of an SRAM cell that includes the impact of NBTI can be calculated as

$$P_{\mathrm{fail}}(t_{\mathrm{age}}) = \int I(\boldsymbol{x}|t_{\mathrm{age}}) P(\boldsymbol{x}) d\boldsymbol{x}. \tag{14}$$

Here, $\boldsymbol{x}$ is a vector of random variable and t_{age} is a chip age. $I(\boldsymbol{x}|t_{\mathrm{age}})$ is an indicator function that returns "1" when the variability $\boldsymbol{x}$ causes a malfunction of the circuit whose age is given by t_{age}. From the above discussion, all the random variables now follow normal distributions. Hence, $\boldsymbol{x}$ follows a multi-dimensional standard normal distribution.

Let us see how the failure region in the variability space

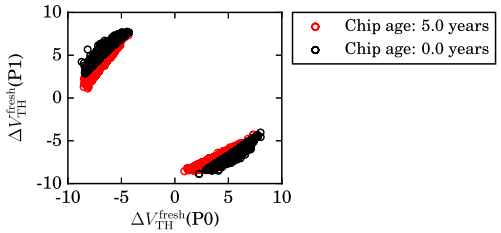


Fig. 4 Temporal change of failure samples in the variability space of a fresh V_{TH}.

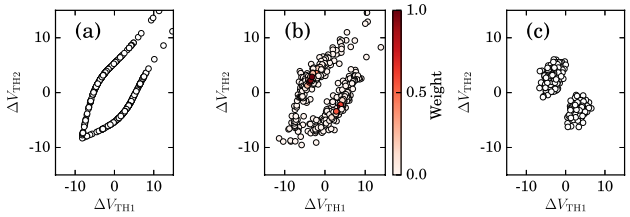


Fig. 5 An example of particle filter based failure region tracking. (a) Particles after initialization step, (b) after prediction and weight calculation steps and (c) after resampling step.

Algorithm 1 Proposed calculation flow

1: (1) **Initial sample selection**
2: **for each** chip age **in** list of ages **do**
3: **repeat**
4: (2) **Prediction**
5: (3) **Measurement**
6: (4) **Resampling**
7: **until** Sufficient convergence of the particle density is achieved
8: (5) **Importance sampling:** construct the alternative distribution and calculate the failure probability at the current aging time step.
9: **end for**

of ΔV_{TH}^{fresh} changes as the chip age increases. Sample points in Fig. 4 show ΔV_{TH}^{fresh} of failure cells. Here, we consider the variability of two pMOS transistors only for simplicity. The black markers show those of fresh cells while the red markers show those of 5-year-old cells. In this example, an SRAM cell who has negative read noise margin is labeled as a failure cell. We notice that there is no drastic change in ΔV_{TH}^{fresh} between fresh and the aged cells. In the proposed method, the alternative distribution are "reused" by continuously modifying it among the multiple failure probability calculations along aging time steps. It eliminates time consuming initial failure-region explorations conducted repeatedly for different chip ages.

The following steps are the overview of the proposed method. Algorithm 1 summarizes the calculation flow.

Initialization Initialize particles positions. The variability space is explored in the radial direction to find the failure regions that are close to the origin. Then, the particles are generated around the failure regions ((1) in Algorithm 1).

Particle filter (first stage) The locations of the particles are then iteratively adjusted so that they best fit the density of an optimal alternative distribution ((2) to (4)).

Importance sampling (second stage) A large number of random samples is generated according to the density of the particles and the failure probability is calculated accurately in (5).

In the failure probability calculation of the second or later aging time steps, the initialization step is skipped. Instead, the particles are copied from the previous calculation and the steps (2) to (4) are conducted so that the positions of the particles are adjusted according to its age.

4.3 Detailed Procedures of the Proposed Method

(1) Initial sample selection

Random samples on the surface of a D-dimensional unit sphere are generated. The boundary of the failure region is searched using bi-section algorithm along the radial directions of the generated random samples. Candidates of initial particles $\{\boldsymbol{x}^{(0,i)}; i = 1, 2, \cdots, N\}$ are allocated near the boundary as shown in Fig. 5(a). Here, N is the total number of particles and $\boldsymbol{x}^{(t,i)}$ is the i-th particle at t-th iteration. Note again that the initialization step is conducted only once. In the failure probability calculations of the succeeding aging time steps, particles are copied from the previous calculations.

Steps from "prediction" to "resampling" are repeated to let the particles to follow the optimal alternative distribution. In our experiment, five to ten times of repetitions are sufficient to achieve convergence in the estimated probability.

(2) Prediction step

The candidate particles at the next iteration $\{\widehat{\boldsymbol{x}}^{(t+1,i)}; i = 1, 2, \cdots, N\}$ are drawn from a mixture of normal distributions:

$$\widehat{\boldsymbol{x}}^{(t+1,i)} \sim \frac{1}{N}\sum_{j=1}^{N} \mathcal{N}_{\mathcal{D}}(\boldsymbol{x}^{(t+1,i)}|\boldsymbol{x}^{(t,j)}, \boldsymbol{\sigma}). \tag{15}$$

Here, $\mathcal{N}_{\mathcal{D}}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\sigma})$ is a D-dimensional normal distribution whose PDF is given by

$$\mathcal{N}_{\mathcal{D}}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{1}{\sqrt{2\pi^D|\boldsymbol{\sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\mu)^T\boldsymbol{\sigma}^{-1}(\boldsymbol{x}-\mu)\right), \tag{16}$$

where $\boldsymbol{\sigma}$ is a diagonal covariance matrix assuming the "whitening" process has been conducted.

(3) Measurement

For each generated candidate particle $\widehat{\boldsymbol{x}}^{(t+1,i)}$, weight $w^{(t+1,i)}$, which is a scalar value representing the fitness of the particle to the optimal alternative distribution, is calculated:

$$w^{(t+1,i)} = I(\widehat{\boldsymbol{x}}^{(t+1,i)}|t_{\text{age}})P(\widehat{\boldsymbol{x}}^{(t+1,i)}). \quad (17)$$

Here, $P(\boldsymbol{x})$ is the probability that the sample $\boldsymbol{x}$ is observed. In our case, it is represented as the PDF of the D-dimensional standard normal distribution which is given by (16) with an identity covariance matrix.

For the computation of $I(\boldsymbol{x}|t_{\text{age}})$, transistor level simulations are required. N samples need to be simulated for the weight calculations of all particles. We reduce the number of simulations with the help of the SVM-based classifier. First, K training examples are randomly selected from N samples and give labels to them using transistor level simulations. Then, the classifier is trained using the K training examples. Finally, the remaining $N - K$ samples are classified using the trained classifier. The number of transistor level simulations can be reduced from N to K.

SVM-based classifier was first applied to a failure probability calculation in [20]. As we have seen, the drawback of the naive MC is that very few failure samples are drawn from the original PDF. Therefore, the authors of [20] used SVM-based classifier as a blockade so that they can skip transistor level simulations for the samples that obviously fall outside the failure region.

The difference between our approach and [20] is that we combine the classifier with the importance sampling. In the context of a failure probability calculation, the variability space is not equally important, i.e., $I(\boldsymbol{x}) \cdot P(\boldsymbol{x})$ represents the importance of the corresponding region. Misclassification of samples that are rare in terms of the alternative distribution, have almost no contribution to the failure probability. We call the number of misclassification weighted by $I(\boldsymbol{x}) \cdot P(\boldsymbol{x})$ as "effective misclassification rate."

In our proposal, the SVM-based classifier is trained using the samples drawn from the alternative distribution, with which the samples are concentrated about the regions that have large weight. Hence, the performance of the classifier can be improved with a smaller number of training samples as compared to the case that the training is conducted by using uniformly generated samples.

The computational cost required to train the classifier increases by $O(n^2)$ where the number of training samples is n. To cover the failure regions of fresh and aged cells with a single classifier, a large number of training samples is required and eventually the training time of the classifier becomes unignorable. In our implementation, the old training samples are discarded and the binary classifier is newly trained for each aging time step to save the number of training samples and the time required for the training. This approach works well because of the monotonic nature of aging — the particles at the current aging time step are unlikely to revisit the regions that have been previously explored.

In order to construct a non-linear classification model, we use a polynomial transform of the variability vector $\boldsymbol{x}$ as feature quantity $\boldsymbol{f}$ in (6). For example, for a two-dimensional input vector $[x_1, x_2]$, the feature vector is $[1, x_1, x_2, x_1x_2, x_1^2, x_2^2]$ when degree of the polynomial transform D_{poly} is quadratic.

Algorithm 2 Pass or fail label estimation.

1: Let $\boldsymbol{X}$ be the particles whose pass or fail labels need to be determined and $\boldsymbol{X}_{\text{train}}$ be the training data for the binary classifier.
2: Let Acc be the accuracy of the classifier. Here, accuracy is calculated as the number of samples whose labels match with those obtained using a SPICE simulator, divided by the total number of samples.
3: **while** True **do**
4: Randomly select K samples from $\boldsymbol{X}$. These samples are then given random perturbations drawn from a Gaussian distribution to obtain a test data $\boldsymbol{X}_{\text{test}}$.
5: Obtain pass or fail labels for $\boldsymbol{X}_{\text{test}}$ using a SPICE simulator.
6: **if** $\boldsymbol{X}_{\text{train}}$ is empty **then**
7: Initialize the accuracy of the classifier (Acc) to be zero.
8: **else**
9: Calculate the accuracy of the classifier on the test data set $\boldsymbol{X}_{\text{test}}$ and Acc is initialized as the calculated accuracy.
10: **end if**
11: **if** Acc is greater than θ **then**
12: Break the while loop.
13: **else**
14: Append the test data to the training data: $\boldsymbol{X}_{\text{train}} = \boldsymbol{X}_{\text{train}} \cup \boldsymbol{X}_{\text{test}}$.
15: Retrain the classifier with the extended training data $\boldsymbol{X}_{\text{train}}$.
16: **end if**
17: **end while**
18: Obtain the pass or fail labels of $\boldsymbol{X}$ using the trained classifier.

The accuracy of the classifier at every aging time is tested using a small subset of samples extracted from the complete set of samples whose pass or fail labels are to be obtained. The incremental refinement is conducted until the sufficient classification accuracy can be achieved. Details of the pass or fail label estimation with the aid of the binary classifier are summarized in Algorithm 2.

Samples that are close to the separating hyper-plane, i.e. colored region in Fig. 2, may be misclassified depending on the accuracy of the classifier. Such samples should be better classified by using transistor-level simulations. However, the weights of particles do not have direct impact on the failure probability calculation. Instead, it only affects to the estimation of the optimal alternative distribution and the efficiency of the importance sampling. Hence, the approximation of $I(\boldsymbol{x}|t_{\text{age}})$ need not to be very accurate in this step. In our implementation, the target accuracy θ in Algorithm 2 and the size of the test data K are 0.98 and 1000, respectively. Note that the number of samples required to train the classifier is sufficiently small and hence the total calculation time can be greatly reduced.

Figure 5(b) shows particles after the prediction and measurement steps. The marker color represents the weight of each particle. We notice that the particles located closer to the origin, where the failure is more likely to occur, have larger weights.

(4) Resampling

Particles at the next iteration step ($\{\widehat{\boldsymbol{x}}^{(t+1,i)}; i = 1, 2, \cdots, N\}$) are randomly selected from the particles $\widehat{\boldsymbol{x}}^{(t+1,i)}$ according to the probability in proportion to their weights. An example result of the resampling step is shown in Fig. 5(c).

While particle filters drastically speed up the estima-

tion of the alternative distribution, we have to consider a degeneration problem of particles. In the failure probability calculation of an SRAM cell, there are two major failure regions because of its symmetric structure. Small difference of the particle weight can make particles to concentrate on one of the two regions as the number of iterations increases. This leads to underestimation of the failure probability, which is undesirable. In the proposed method, we utilize multiple particle filters. The resampling is limited within each particle filter in order to avoid the degeneration problem.

In the example in Fig. 5(c), the two major failure regions are tracked by different particle filters.

(5) Importance sampling

Finally, in the second stage, the failure probability is calculated using an importance sampling. In order to optimize the alternative distribution, the outcome of the previous stage is used. Specifically, the distribution of the particles in step (4) is very close to the optimal distribution, hence it is approximated as

$$\widehat{Q}(\boldsymbol{x}) \approx \frac{1}{N}\sum_{i=1}^{N} \mathcal{N}_{\mathcal{D}}(\boldsymbol{x}|\boldsymbol{x}^{(t,i)}, \boldsymbol{\sigma}). \tag{18}$$

Then, the failure probability is calculated using random samples $\{\boldsymbol{x}_{\mathrm{IS}}^k, k = 1, 2, \ldots, N_{\mathrm{IS}}\}$ drawn from $\widehat{Q}(\boldsymbol{x})$ as follows:

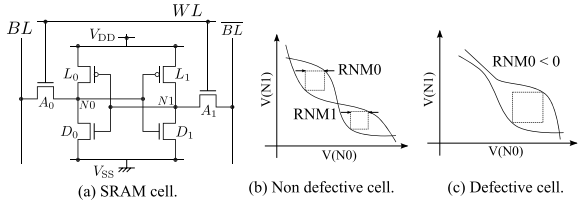
$$P_{\mathrm{fail}} \approx \frac{1}{N_{\mathrm{IS}}}\sum_{k=1}^{N_{\mathrm{IS}}} I(\boldsymbol{x}_{\mathrm{IS}}^k|t_{\mathrm{age}})P(\boldsymbol{x}_{\mathrm{IS}}^k)/\widehat{Q}(\boldsymbol{x}_{\mathrm{IS}}^k). \tag{19}$$

Here, N_{IS} is the number of random samples used for the approximation. The calculation of the indicator function is again needed in the evaluation of $I(\boldsymbol{x}_{\mathrm{IS}}^k|t_{\mathrm{age}})$. We again use the SVM-based binary classifier to reduce the number of simulations. Contrary to the classification in the first stage, classification accuracy in the second stage directly impacts on the accuracy of the failure probability calculation. Therefore, the samples which lie close to the separating hyperplane go through the transistor-level simulations to obtain correct labels. The simulated samples are used to incrementally train the classifier and to increase the classification performance.

5. Numerical Experiment

5.1 Experimental Setup

Figure 6 shows the circuit schematic of an SRAM cell. In the experiment, failure samples are defined as samples which have negative read noise margin (RNM). RNM is a stability measure of the cell, which can be computed as the maximum size of square embedded within the opening of the butterfly curve [21] of the cell. Figures 6(b) and (c) show two examples for defective and non-defective cells. The mismatch of driving abilities among transistors results in negative noise margin, which causes the read failure.



(a) SRAM cell. (b) Non defective cell. (c) Defective cell.

Fig. 6 (a) The schematics of the SRAM cell and (b) examples of static noise margin for a non-defective and (c) a defective cell.

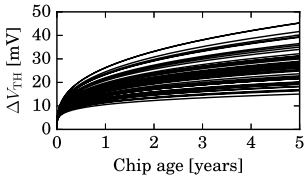


Fig. 7 Examples of NBTI-induced V_{TH} shift assumed in this experiment.

Table 1 Experimental conditions.

	Load (L_i)	Driver (D_i)	Access (A_i)
$A_{V_{\mathrm{TH}}}$ [mV·nm]	5×10^2		
Channel length [nm]	16		
Channel width [nm]	30	50	30
k [V/sec]	2×10^{-3}		
μ_n	−1.3		
σ_n	0.1		

In the experiment, the 16 nm high-performance model from predictive technology model (PTM) [22] is used as a transistor model. Long-term V_{TH} degradation is predicted using the following model that is slightly modified from the original RD-model in (7):

$$\Delta V_{\mathrm{TH}}^{\mathrm{NBTI}} = k \cdot \left(C_t \cdot t_{\mathrm{age}}\right)^n, \tag{20}$$

where t_{age} is a chip age in year and C_t is a constant to adjust the time scale. The model parameters are selected so that approximately 20 mV to 30 mV of V_{TH} shift is observed in the transistors of 5-years old. In this particular case, k and C_t are assumed to be 2×10^3 and 2×10^5, respectively. The logarithm of the power-law exponent ($n_{\log}$) is assumed to follow a normal distribution (9), where μ_n and σ_n are set to −1.3 and 0.1, respectively. Figure 7 shows example of V_{TH} shifts of 50 pMOS transistors assumed in this experiment. Other circuit parameters such as gate length and width are summarized in Table 1. In order to see the temporal change of the failure probability at the early ages, where V_{TH} rapidly increases, the aging time step is selected as follows: t_{age} = 0, 0.01, 0.03, 0.05, 0.1, 0.3, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0 years. The degree of the transform polynomial for the SVM-based classifier, D_{poly}, is four and the number of particle filters is set to ten.

5.2 Experimental Results

We first compare the proposed method with one of the state-of-the-art methods proposed in [6]. Note that, in this experiment, the failure probability at a single aging time step is

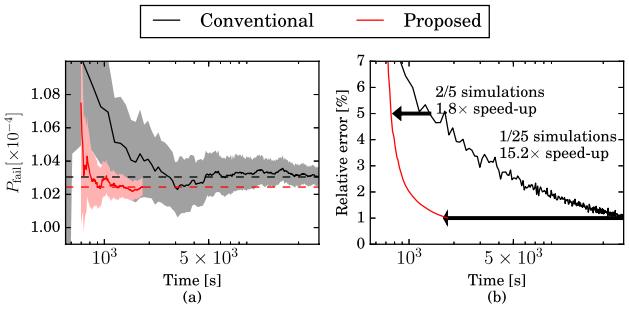


Fig. 8 The comparison of the proposed and the conventional [6] methods. (a) The relationship between the calculated failure probability and the calculation time required. (b) The relationship between the relative error and the calculation time required.

calculated to see the effectiveness of the two-stage MC and the SVM-based binary classifier. Figure 8(a) shows the calculated failure probability of the SRAM cell and required calculation time. The calculation time was measured on a Linux workstation with 6-core Xeon X5670 processor operated at 2.93 GHz. All of the 6-cores are used to accelerate the calculation. In Fig. 8(a), the filled regions represent the 95% confidence intervals. We can see that the proposed method converges faster than the conventional method. Figure 8(b) shows relative error as a function of calculation time. The relative error is defined as the ratio of the 95% confidence interval to the calculated failure probability. In our experimental setup, the proposed method reduced the number of simulation runs into 40% of that of the conventional method [6] to achieve relative error of 5%. The proposed method required about 760 seconds to obtain that accuracy. This includes both training time of the classifier and classification time. The conventional method required about 1400 seconds to achieve an equal accuracy, which corresponds to about 1.8× speed-up. When the acceptable error is small, the difference in the calculation time becomes large. For example, the proposed method can achieve 15.2× speed up over the conventional method when the permissible error is set to 1%.

We then calculate the temporal change of the failure probability with the proposed method and summarize the result in Fig. 9(a). As a comparison, the result of the conventional method is shown in Fig. 9(b). The permissible error is set to 5%. We can see that the results of both methods are almost equal, from which we can confirm the correctness of the proposed method. The total calculation time required to obtain Fig. 9 is about 5700 seconds for the proposed method while 76700 seconds for the conventional method. The proposed method, hence, achieves 13.4× speed up compared to the conventional method. The magnitude of the speed up is increased from the comparison in Fig. 8 because the comparison in Fig. 9 includes the effect of the particle reusing. In the conventional method, the alternative distribution is constructed from scratch at each aging time step while in the proposed method, the construction is conducted only once, which further reduced the total calculation time.

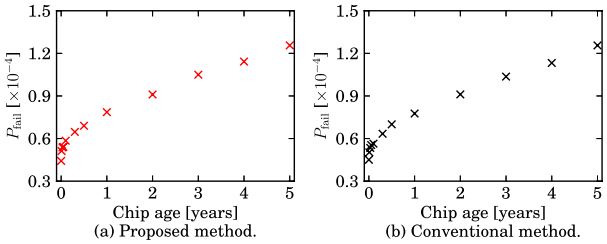


Fig. 9 The temporal change in the failure probability.

6. Conclusion

In this paper, we proposed a novel method to efficiently calculate the failure probability of an SRAM cell that can take the impact of NBTI-induced device degradation into account. To see the temporal change of the failure probability at different device ages, multiple failure probability calculations are required by changing threshold voltages of transistor at an aging time step. Considering the gradual V_{TH} change due to aging, we proposed a method that utilizes particle filter to keep track the change of the near optimal alternative distribution for importance sampling. With this idea, time consuming repetitive explorations in the variability space has been eliminated. Combined with a binary classifier and two-stage MC approach to further reduce the calculation time, the proposed method achieved 13.4× speed-up over one of the state-of-the-art method [6]. With the aid of the proposed method, circuit designers can efficiently see the impact of device degradation. Reliability of highly scaled LSIs can be improved, contributing to broader adoption of LSIs.

Acknowledgment

This work was partially supported by a Grant-in-Aid for JSPS Fellows and JSPS KAKENHI Grant No. 26280014. The authors also acknowledge support from VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc.

References

[1] A.J. Bhavnagarwala, X. Tang, and J.D. Meindl, "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," IEEE J. Solid-State Circuits, 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," Proc. 43rd Annual Conference on Design Automation, DAC'06, pp.69–72, 2006.

[3] J. Jaffari and M. Anis, "Adaptive sampling for efficient failure probability analysis of SRAM cells," Proc. 2009 International Conference on Computer-Aided Design, ICCAD'09, pp.623–630, 2009.

[4] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan, "Breaking the simulation barrier: SRAM evaluation through norm minimization," 2008 IEEE/ACM International Conference on Computer-Aided Design, pp.322–329, 2008.

[5] J.H. Stathis and S. Zafar, "The negative bias temperature instability in MOS devices: A review," Microelectron. Reliab., vol.46, no.2-4,

pp.270–286, 2006.
[6] K. Katayama, S. Hagiwara, H. Tsutsui, H. Ochi, and T. Sato, "Sequential importance sampling for low-probability and high-dimensional SRAM yield analysis," 2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp.703–708, 2010.
[7] E. Maricau and G. Gielen, "Stochastic circuit reliability analysis," 2011 Design, Automation & Test in Europe, pp.1–6, 2011.
[8] E. Maricau and G. Gielen, "Efficient variability-aware NBTI and hot carrier circuit reliability analysis," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.29, no.12, pp.1884–1893, 2010.
[9] K. Kang, H. Kufluoglu, K. Roy, and M.A. Alam, "Impact of negative-bias temperature instability in nanoscale SRAM array: Modeling and analysis," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.26, no.10, pp.1770–1781, 2007.
[10] M. Rana and R. Canal, "SSFB: A highly-efficient and scalable simulation reduction technique for SRAM yield analysis," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2014, pp.1–6, 2014.
[11] C. Dong and X. Li, "Efficient SRAM failure rate prediction via Gibbs sampling," Proc. 48th Design Automation Conference, DAC'11, pp.200–205, 2011.
[12] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," IEE Proc. F Radar Signal Process., vol.140, no.2, pp.107–113, 1993.
[13] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," J. Comput. Graph. Stat., vol.5, no.1, pp.1–25, 1996.
[14] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol.20, no.3, pp.273–297, 1995.
[15] K.B. Sutaria, J.B. Velamala, C.H. Kim, T. Sato, and Y. Cao, "Aging statistics based on trapping/detrapping: Compact modeling and silicon validation," IEEE Trans. Device Matel. Rel., vol.14, no.2, pp.607–615, 2014.
[16] T. Sato, T. Kozaki, T. Uezono, H. Tsutsui, and H. Ochi, "A device array for efficient bias-temperature instability measurements," 2011 Proc. European Solid-State Device Research Conference (ESSDERC), pp.143–146, 2011.
[17] T. Grasser, B. Kaczer, W. Goes, H. Reisinger, T. Aichinger, P. Hehenberger, P.-J. Wagner, F. Schanovsky, J. Franco, M.T. Luque, and M. Nelhiebel, "The paradigm shift in understanding the bias temperature instability: from reaction-diffusion to switching oxide traps," IEEE Trans. Electron Devices, vol.58, no.11, pp.3652–3666, 2011.
[18] H. Awano, M. Hiromoto, and T. Sato, "Variability in device degradations: Statistical observation of NBTI for 3996 transistors," 2014 44th European Solid State Device Research Conference (ESSDERC), pp.218–221, 2014.
[19] T. Mizutani, A. Kumar, and T. Hiramoto, "Analysis of transistor characteristics in distribution tails beyond $\pm 5.4\sigma$ of 11 billion transistors," 2013 IEEE International Electron Devices Meeting, pp.33.3.1–33.3.4, 2013.
[20] A. Singhee and R.A. Rutenbar, "Statistical blockade: Very fast statistical simulation and modeling of rare circuit events and its application to memory design," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.28, no.8, pp.1176–1189, 2009.
[21] E. Seevinck, F.J. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," IEEE J. Solid-State Circuits, vol.22, no.5, pp.748–754, 1987.
[22] Nanoscale Integration and Modeling (NIMO) Group, ASU, "Predictive technology model (PTM)." http://ptm.asu.edu/

Hiromitsu Awano received his B.E. degree in Informatics and his master degree in Communications and Computer Engineering from Kyoto University in 2010 and 2012, respectively. Presently, he is a doctor course student at Department of Communications and Computer Engineering, Kyoto University. He is a research fellow of japan society for the promotion of science and a student member of IPSJ and IEEE.

Masayuki Hiromoto received B.E. degree in Electrical and Electronic Engineering and M.Sc. and Ph.D. degrees in Communications and Computer Engineering from Kyoto University in 2006, 2007, and 2009 respectively. He was a JSPS research fellow from 2009 to 2010, and with Panasonic Corp. from 2010 to 2013. In 2013, he joined the Graduate School of Informatics, Kyoto University, where he is currently an assistant professor. His research interests include VLSI design methodology, image processing and pattern recognition. He is a member of IEEE and 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.