SRAM Parametric Failure Analysis

Jian Wang[1], Soner Yaldiz[2], Xin Li[2], Lawrence T. Pileggi[2]

[1] PDF Solutions, Inc., San Jose, CA 95110

[2] Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

[1] jian.wang@pdf.com, [2] {syaldiz, xinli, pileggi}@ece.cmu.edu

ABSTRACT

With aggressive technology scaling, SRAM design has been seriously challenged by the difficulties in analyzing rare failure events. In this paper we propose to create statistical performance models with accuracy sufficient to facilitate probability extraction for SRAM parametric failures. A piecewise modeling technique is first proposed to capture the performance metrics over the large variation space. A controlled sampling scheme and a nested Monte Carlo analysis method are then applied for the failure probability extraction at cell-level and array-level respectively. Our 65nm SRAM example demonstrates that by combining the piecewise model and the fast probability extraction methods, we have significantly accelerated the SRAM failure analysis.

Categories and Subject Descriptors

B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids

General Terms: Algorithms, Reliability

Keywords: SRAM, Parametric Failure, Failure Probability Estimation, Response Surface Model

1. INTRODUCTION

As the most commonly used embedded memory of modern on-chip systems, SRAM plays a crucial role in defining the system performance [1]. For maximum storage density, SRAM bit-cells have always been designed to use near-minimal devices in a given technology node. Such tight layout footprints make SRAM cells extremely vulnerable to the performance degradation and failures caused by random dopant fluctuation (RDF), which is inversely proportional to the layout area [2,3].

The RDF induced variations are spatially independent in nature. It follows that a rare cell-level failure caused by such variations can become quite significant for a system with many replicated cells. With the aggressive technology scaling, such local random variations are becoming more dominant. Therefore SRAM cells have to be carefully designed to provide an exceedingly low failure rate, such that the functionality is maintained for the system which contains thousands, or even millions of such identical cells [4]. How to probe such rare failure events has become a serious challenge for SRAM design and analysis.

DAC'09, July 26-31, 2009, San Francisco, California, USA

A widely used approach to study such a probability problem is the Monte Carlo method [5], which provides statistical estimations based on experiments performed at randomly selected samples in the variation space. To estimate the probability of the rare failure events in SRAM circuits, however, the Monte Carlo method typically requires millions, or even billions of samples to reach reasonable accuracy [6,7]. Such prohibitive cost severely limits the application of the Monte Carlo method in SRAM analysis. To alleviate this problem, several improvements have been proposed by integrating some controlling scheme in the sampling process, such as Latin hypercube sampling [8] or low-discrepancy sampling [7]. These methods claim to offer comparable accuracy to Monte Carlo methods while running up to a hundred times faster. Nevertheless, as demonstrated later in this paper, such an enhancement is still insufficient to enable SRAM failure analysis.

An alternative avenue to attack the problem is by modeling the stability metric as a known distribution and calculating the failure probability explicitly. For example, Gaussian distributions, non-central F distributions, or generalized Pareto distributions are among the commonly-used forms for approximating the stability metrics [9-12]. The accuracy of these methods, however, heavily relies on the validity of their assumptions, which unfortunately, are often questionable for nanoscale SRAM circuits. As pointed out by [6] and [13], even when the center portion of the stability metric distribution closely matches the presumed type, the tail part, where the failures usually happen, often deviates from the assumed form due to the increased nonlinearity in those regions.

In this paper we address the problem of SRAM parametric failure analysis with two novel techniques. Firstly, a response surface modeling (RSM) approach is applied to reduce the cost of transistor-level simulations. The critical problem here is how to accurately model the performance metric over a large variation space. We designed a piecewise approach, which by adaptively partitioning the variation space, renders a set of models covering the entire space while providing superb accuracy in regions critical for failure classification. Secondly, based on the statistical performance models, we explicitly identify the failure regions and apply a controlled sampling scheme to better probe those areas. Our approach effectively reduces the sample size without any assumptions on the performance metric distribution. By combining the model-based evaluation and the efficient sample-allocation, we significantly accelerate SRAM failure analysis, as compared to the traditional Monte Carlo approach.

The remainder of the paper is organized as follows. In Section 2, we define the SRAM failure analysis problem and review some background techniques. Section 3 presents our methods for fast

probability extraction, i.e. the piecewise modeling and the controlled sampling techniques. We then escalate the problem in Section 4 and discuss how to estimate the failure probability for a SRAM array. Some results from a 65nm design are then shown in the next section, followed by our conclusions in Section 6.

2. BACKGROUND

2.1 SRAM Parametric Failure Analysis

To ensure proper functionality over the process variations, we focus on *stability margins* that are defined for SRAM cells as measures of robustness under different operating scenarios. Violation of a stability margin specification at certain process point is referred to as a *parametric failure* (or in short, a *failure*) in this paper. It should be noted, however, nothing would preclude us from considering other SRAM performance metrics for failure analysis by using the proposed techniques.

Assume that the process variations are described by the vector $x \in \Re^k$ of k independent random variables, which include both the global process variables and the local process variables from all transistors in the SRAM cell. The distribution of x is defined by its probability density function (PDF) $\rho(x)$. We choose a metric $S(x)$ and a specification S_{SPEC} to test the stability of the SRAM cell. $S(x) \geq S_{SPEC}$ signifies that the cell is stable at the process point x, and vice versa. Thereby we define the indicator function:

$$\mathcal{I}(x) \equiv \begin{cases} 1 & S(x) < S_{SPEC} \\ 0 & S(x) \geq S_{SPEC} \end{cases} \tag{1}$$

where $\mathcal{I}(x) = 1$ indicates the failure event. One very important problem in the SRAM stability analysis is to find the parametric yield loss, i.e. the failure probability due to the process variations:

$$P \equiv \mathcal{P}\left(\mathcal{I}(x) = 1\right) \tag{2}$$

The most commonly used approach for probability estimation is the Monte Carlo (MC) method [5]. Traditional Monte Carlo approach consists of three steps: (a) a total number of N samples, $x^{\{i\}}$ $(i = 1, \cdots, N)$, are randomly selected in the process variation space according to the distribution $\rho(x)$; (b) the interested stability metric is tested at all the sample points; (c) the results, $S(x^{\{i\}})$ and $\mathcal{I}(x^{\{i\}})$, are aggregated for estimation.

The failure probability can be estimated by averaging the indicator function:

$$P_{MC} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{I}(x^{\{i\}}) \tag{3}$$

Due to the randomness in the sampling, the estimated probability varies with different set of samples with a variance as [5]:

$$\sigma^2[P_{MC}] = \frac{P(1-P)}{N} \cong \frac{P_{MC}(1-P_{MC})}{N-1} \tag{4}$$

where the operator "$\cong$" means "estimated by" (since the actual probability P is not known a priori).

As previously discussed, when analyzing the failure events in SRAM cells, we expect the probability to be extremely small. In such case, a meaningful estimation requires $\sigma[P_{MC}]$ to be sufficiently smaller than the actual probability. We define the *confidence ratio* to qualify the "accuracy" of the estimation:

$$R_C \equiv \frac{\sigma[P_{MC}]}{P} = \sqrt{\frac{1-P}{NP}} \approx \sqrt{\frac{1}{NP}} \tag{5}$$

Clearly, if we want to retain a low R_C while P is very small, the sample count N has to be extremely large. This point is illustrated with a simple example in Fig. 1, where we use the Monte Carlo method to study the tail of a standard normal distribution and obtain the probability $\mathcal{P}(x > n_\sigma)$ (for convenience we often map an extremely small probability to a normal distribution and represent its location in the unit of σ). The plot shows the number of samples needed to reach a R_C of 0.05. Evidently, the sample count increases intractably when the event being analyzed moves towards the tail of its distribution.

For SRAM cell failure analysis, it is very usual to have failure probabilities beyond 5σ [4,6]. As demonstrated in Fig. 1, the excessive samples required by direct Monte Carlo method are often beyond practical means.

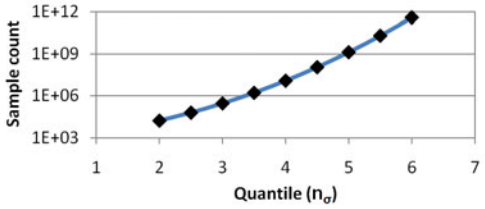


Figure 1. Samples needed by Monte Carlo method

2.2 Importance Sampling

Intuitively, with random sampling and a limited sample size, it is very unlikely that many samples are placed into the region that causes rare failures. This, in turn, renders large estimation error (confidence ratio) in the direct Monte Carlo method. To alleviate this problem, importance sampling (IS) is applied to the SRAM applications [6].

In importance sampling, a biased sample distribution $g(x)$ is introduced to intentionally place more samples into the failure region. After the controlled sampling and experiments, the failure probability is estimated as [14]:

$$P_{IS} = \frac{1}{N}\sum_{i=1}^{N}\left[\mathcal{I}(x^{\{i\}})\frac{\rho(x^{\{i\}})}{g(x^{\{i\}})}\right] \tag{6}$$

Note that compared to (3), the averaging here is weighted to unbias the estimation. The estimation variance can be derived as:

$$\sigma^2[P_{IS}] \cong \frac{\sum_{i=1}^{N}\left[\mathcal{L}(x_{\{i\}})^2 - P_{IS}^2\right]}{N(N-1)} \tag{7}$$

where $\mathcal{L}(x^{\{i\}}) \equiv \mathcal{I}(x^{\{i\}})\rho(x^{\{i\}})/g(x^{\{i\}})$. If $g(x) = \rho(x)$, Equ. (7) converges to Equ. (4). But as $g(x)$ shifts more emphasis onto the failure region, the dispersion in $\mathcal{L}(x^{\{i\}})$ decreases, as well as the variance $\sigma^2[P_{IS}]$.

By placing more samples in the failure region, importance sampling is expected to provide a much tighter estimation, compared to direct Monte Carlo method with the same number of random samples. One key issue, however, is how to design the new distribution $g(x)$. The authors of [6] attempt to first locate

the region of failures by uniformly sampling the variation space, and then construct $g(x)$ based on such information. This causes two problems, however: (a) searching the process space with uniform samples can be very ineffective, especially when the failure events are rare or the space dimension is high [14]; (b) even the sample number is effectively reduced by importance sampling, the simulation cost for evaluating the samples could still be very expensive, as demonstrated later in this paper.

3. FAST ESTIMATION OF SRAM CELL FAILURE PROBABILITY

In view of the challenges in the SRAM stability analysis, we propose to build accurate statistical performance models that facilitate the probability extraction. A controlled sampling scheme is then performed for better failure region coverage and sample reduction. This section describes these two techniques.

3.1 Performance Modeling of Stability Metrics

There are two motivations for introducing response surface model (RSM) into the SRAM stability analysis problem. First, with the model it is possible to accurately locate the region of failures and apply a precisely-designed controlled sampling scheme. Second, the RSM hides the internal workings of the circuit and therefore can greatly accelerate the sample evaluation process.

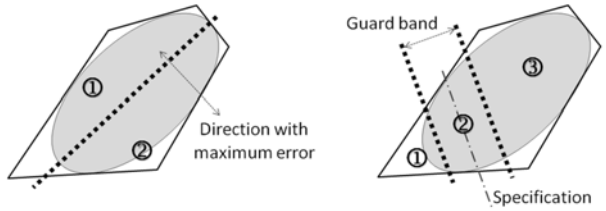
With the increasing process variations, however, the circuit performance exhibits stronger nonlinear effects and is very difficult to be accurately captured. Furthermore, particularly in the SRAM applications, the cell stability failures have to be controlled at an exceedingly low level, which demands the scope to be extended further into the tails of the parameter distributions.

Facing these challenges, we propose to partition the variation space and create piecewise response surface models for the stability metrics. The authors of [15] developed a systematic approach to partition a parameter space and create piecewise models. Although initially proposed for analog macromodeling, with proper modifications this method is applicable to SRAM failure analysis as well.

In summary, we use the upper and lower bounds of the process parameters to define the initial region as a hypercube. This space is then adaptively divided into smaller pieces: (a) the inscribed ellipsoid of the current polytope (a hypercube in the first step) is first found by convex optimization; (b) the local space is sampled by a Design-of-Experiment (DoE) approach, and at each sample point the SRAM cell is simulated for its stability margin; (c) with the samples, a linear local model is created and the modeling error is evaluated to decide if further partitioning is necessary; (d) partitions with large errors will be further divided and the above steps are recursively applied. After the partitioning, the stability margin is modeled as a collection of the local models in all final partitions. More details of the partitioning formulation can be found in [15] and are neglected due to space limitations here.

Unlike [15], where a simple error-based criterion is applied to decide the partitioning direction, here we employ a set of criteria to better suit the particular probability extraction problem. Specifically, since our eventual goal is to estimate the failure probability, the modeling error does not need to be uniformly controlled over the entire variation space. For regions where the value of the stability metric is close to the predefined specification, high accuracy is required because it directly affects the correctness of the failure identification. For the remaining regions, the model does not need to be as accurate, as long as the modeling error does not reverse the specification pass/fail status at that process point.

For this reason, we selectively apply a response-based criterion and an error-based criterion in different stages of the partitioning process. Both methods are illustrated in Fig. 2. The error criterion first finds the direction with the maximum modeling error. A hyperplane passing the ellipsoid center is then selected to be orthogonal to that direction and is used to divide the current partition into two pieces. The purpose of such partitioning is to reduce the size the local space and to increase the model accuracy. Alternatively, the response criterion constructs the contour of the specification, which is a hyperplane since the local model is linear. With some guard band added to both sides of the contour, the current polytope is divided into three parts. The purpose of this is to isolate the region requiring higher accuracy.



(a) Error-based (b) Response-based

Figure 2. Partitioning criteria

Before the partitioning, we define a larger error tolerance ε_H and a smaller one ε_L. Intuitively, the partitioning always starts with the error criterion approach, until the higher tolerance ε_H is met. Then, the response-based approach is applied to divide the current polytope into three pieces. For the two pieces away from the specification, we stop further partitioning. While for the center piece, the partitioning carries on using again the error-based approach but with the lower tolerance ε_L as target. By this flow, we identify the regions where the stability metric is close to the specification, and create local models with higher accuracy in those regions. For the rest of the parameter space, the accuracy requirement is relaxed to reduce the modeling cost.

3.2 Model-based Probability Extraction

After the piecewise modeling, we have the stability metric represented as a linear model in each partition. We can then estimate the failure probability of the SRAM cell in three steps. Firstly, we isolate the critical partitions that contain failure regions. This is conveniently done by solving a linear programming problem to find the worst-case stability metric value within each polytope. Secondly, in each critical partition, we construct a new sample distribution to provide better coverage in the failure region and estimate the failure probability with importance sampling. Finally, we aggregate the results from all partitions and gradually append more samples until a target estimation confidence is achieved.

In each critical partition, we construct a new sample distribution as illustrated in Fig. 3. In such a partition, the specification contour $S(x) = S_{SPEC}$ defines a hyperplane. Therefore, very similar to that in the space partitioning process, we can solve a

convex optimization problem to obtain an ellipsoid $\Phi = \{Ex + d \mid \|x\|_2 \le 1\} \subseteq \Re^k$ that approximates the shape of the failure region [16]. As the sample distribution for the importance sampling, we construct a standard multivariate Gaussian distribution residing in the space spanned by the ellipsoid axes. If observed from the original parameter space, the new distribution is centered at the ellipsoid center *d*, and along each ellipsoid axis it presents an independent normal distribution whose standard deviation is proportional to the corresponding axis length. The PDF of the new sample distribution is given as:

$$g(x) = \frac{1}{\left(\sqrt{2\pi}\right)^k \sqrt{|\Lambda|}} \exp\left(-\frac{1}{2}(x-d)^T E^{-1}(x-d)\right) \quad (8)$$

where Λ is a diagonal matrix of the eigenvalues of E.

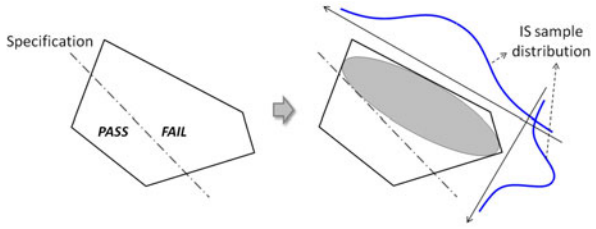


Figure 3. Defining biased sample distribution

We now can apply the importance sampling technique to estimate the failure probability. The samples are generated according to the new distribution $g(x)$ and are evaluated by the RSM. From the samples, we can estimate the failure probability by Equ. (6), and the estimation variance by Equ. (7). When all the partitions are processed, we aggregate the results and obtain the overall failure probability and estimation variance as:

$$P_{cell} = \sum_i P_{IS}^{(i)} \ , \ \sigma^2[P_{cell}] = \sum_i \sigma^2[P_{IS}^{(i)}] \quad (9)$$

where the bracketed superscripts indicate the corresponding model partition.

In addition, due to the *sample superimposability* of the Monte Carlo method [5], it is unnecessary to apply all the samples at once. Instead, a better strategy is to first use a small amount of samples and then gradually append more until a predefined objective is met. In our implementation, we define a target R_C and first use a small number of samples in each partition. Next we select a partition which potentially provides the maximum estimation variance drop. Additional samples are appended to the chosen partition to improve the estimation accuracy there. This process is repeated until the pre-defined target is reached.

In summary, the proposed algorithm accelerates the analysis of SRAM cell failure events from three aspects: (a) With the RSM, we can easily identify the failure region and apply importance sampling with a controlled sample distribution. (b) For sample evaluation, response surface model is used in place of the expensive transistor-level simulation. (c) The samples are added in small chunks and the supplement stops as soon as the predefined estimation confidence is reached.

4. ARRAY-LEVEL ESTIMATION

In addition to the cell-level failure probability described in the previous section, it is often of even more interest to know such probability at array-level or system-level. This analysis problem is not trivial since the failures of individual cells are affected by both independent factors (such as local variations) and correlated factors (such as global variations). To solve this problem, we propose a nested Monte Carlo (NMC) method, where the outer level handles the global variations and the inner level handles the local variations.

The process variation information currently provided by IC foundries is usually defined in two levels. The global variations, defined as vector x_G, affect all the cells uniformly, while the local variations, defined as vector x_L, apply to each cell independently. For simplicity we assume that the SRAM array consists of M identical bit-cells without any redundancy (the handling of redundancy will be discussed at the end of this section). The failure event at the array-level, $\mathcal{I}_{sys}(x_G, x_{L[1]}, \ldots, x_{L[M]})$, is defined as the situation that at least one cell fails (variables connected to an individual cell are marked with square brackets in subscript). Our goal here is to find the failure probability P_{sys} of such array failures.

The nested Monte Carlo method can be described as the following steps: (a) We create the piecewise RSM for the stability metric of an individual cell with both the global and local variations as parameters, i.e. $S(x_G, x_{L[i]})$. (b) K samples are generated in the x_G space, according to the global variation distributions. (c) At each sample $x_G^{\{i\}}$, we apply the technique described in the previous section and estimate the cell failure probability $P_{cell}|_{x_G = x_G^{\{i\}}}$. (d) Given point $x_G^{\{i\}}$, the cell failures are only affected by local variations and are independent. Therefore, the array-level failure probability is obtained as:

$$P_{sys}\Big|_{x_G = x_G^{\{i\}}} = 1 - \left(1 - P_{cell}\Big|_{x_G = x_G^{\{i\}}}\right)^M \quad (10)$$

(e) When all the samples are processed, we estimate the array-level failure probability and the estimation variance as:

$$P_{NMC} = \frac{1}{K}\sum_{i=1}^{K}\left(P_{sys}\Big|_{x_G = x_G^{\{i\}}}\right) \quad (11)$$

$$\sigma^2[P_{NMC}] \cong \frac{\sum_{i=1}^{K}\left(M^2\left(P_{cell}\Big|_{x_G = x_G^{\{i\}}}\right)^2 + M^2\sigma^2\left[P_{cell}\Big|_{x_G = x_G^{\{i\}}}\right]\right) - KP_{NMC}^2}{K(K-1)} \quad (12)$$

The nested Monte Carlo method expedites the analysis of SRAM array failures in several ways. Firstly, the RSM is created only once with both the global and local variations as parameters, and there is no need to create the model over again at different global variation points. Secondly, at the inner-level we estimate the cell failure probability with the proposed importance sampling technique, then the failure probability of the array is analytically calculated by Equ. (10). Thirdly, by analyzing the estimator variance in (12), we notice that it is minimized when P_{cell} does not fluctuate much at different x_G points, i.e. the failure is dominated by the local variations. Fortunately, this is what we expect from the trends for the latest technology nodes [4].

As an alternative approach, we can also estimate the upper bound of the array-level failure probability by assuming all cell failures are independent, i.e.,

$$P_{sys} \le 1 - \left(1 - P_{cell}\right)^M \quad (13)$$

Since the local variations are becoming dominant, we expect such an upper bound to be fairly close to the actual value such that can be used as a fast approximation when accuracy is not very critical.

For simplicity we did not include redundancy in the above description. It should be noted, however, by formulating (10) and (13) accordingly, the proposed methods can be easily applied to systems with redundancy and/or ECC.

5. NUMERICAL EXPERIMENTS

In this section we demonstrate the efficiency of the proposed algorithms with a 6T SRAM cell designed in IBM 65nm CMOS process. Three stability metrics are selected for experiments: static noise margin (SNM) [17], read noise margin (RNM) [17] and write margin (WM) [18] (since we do not require any assumption on the metric distribution, other definitions can be used as well). The proposed methodology is implemented in MATLAB, using Spectre as transistor-level simulation engine.

5.1 Model Creation

For each stability metric, we apply the piecewise modeling methodology to create a response surface model, capturing the variation space up to ±6σ for the local variations. For comparison, we also create a linear model over the entire region. To evaluate the model accuracy, we select 10,000 points uniformly distributed in the process space and choose those points that are in the tail part of the stability metric distribution (close to the specification) to compute the modeling error. The average errors and other relevant results are summarized in Table 1.

Evidently, linear template yields significant errors for all three metrics, including SNM and RNM which many believe to be normally distributed. In contrast, the proposed methodology effectively captures the large variation space and provides superior accuracy for all three metrics.

Table 1. Modeling results

Stability metric	*Model template*	*Partition #*	*Total simulation #*	*Runtime (min)*	*Avg. err. (%)*
SNM	Piecewise	13	0.6 k	7.0	1.9
	Linear	1	25	0.1	8.1
RNM	Piecewise	16	0.8 k	10.0	3.4
	Linear	1	27	0.1	12.1
WM	Piecewise	23	0.8 k	11.3	3.4
	Linear	1	19	0.1	8.2

5.2 Cell-level Failure Probability Extraction

When the piecewise models are available, the RSM-based importance sampling technique is applied to estimate the probability of the selected stability failure for an individual SRAM cell. We set the target confidence ratio R_C to 0.1 and run the proposed estimation flow (IS+RSM). The estimation results are listed in Table 2, which also includes the results from RSM-based and simulation-based Monte Carlo runs (MC+RSM, MC+Sim) with similar estimation confidence. Other methods, such as those in [11] or [12], are not included here since the estimation accuracy information is not attainable in those methods. It should be noted, however, that since the sample sizes required by direct Monte Carlo method are in general beyond our computational capacity, some data (in shaded cells) are extrapolated from actual runs with fewer samples. Data that cannot be extrapolated (such as the failure probability from simulation-based Monte Carlo analysis) are left blank in the table.

The results demonstrate that RSM evaluation is over 400 times faster than transistor-level simulation in general, while the importance sampling technique, as compared to direct Monte Carlo analysis, is able to achieve a sample size reduction of over 2,000 fold. By combining these two features, the proposed method accelerates the SRAM failure analysis by 10^5 to 10^8 times, thereby enabling such difficult analyses.

Table 2. Cell-level estimation results

Metric	*Method*	*Sim./eval. #*	*Runtime*	P_{cell}	$\sigma[P_{cell}]$
SNM	IS+RSM	175 k	1.3 min	5.31e-8	5.3e-9
	MC+RSM	1 B	5.1 day	—	5.4e-9
	MC+Sim	1 B	6.6 yr	—	—
RNM	IS+RSM	105 k	2.5 min	1.17e-7	1.2e-8
	MC+RSM	0.7 B	4.0 day	—	1.2e-8
	MC+Sim	0.7 B	4.9 yr	—	—
WM	IS+RSM	60 k	1.1 min	5.48e-9	4.7e-10
	MC+RSM	40 B	0.8 yr	—	5.0e-10
	MC+Sim	40 B	386 yr	—	—

5.3 Array-level Failure Probability Extraction

Next we demonstrate the nested Monte Carlo (NMC) method on a 16 Kb SRAM array. For the inner-level estimation of the cell failure probability, we again set the R_C as 0.1. At the outer-level, we fix the number of the global variation (x_G) samples to be 100, which renders sufficiently accurate results in our experiments. For comparison, we calculate the sample number needed by direct Monte Carlo method to achieve similar accuracy and project its runtime (based on RSM evaluations). The results are summarized in Table 3. Again extrapolated data are in shaded cells.

Table 3. Array-level estimation results

Metric	*Method*	x_G #	*RSM eval. #*	*Runtime*	P_{sys}	$\sigma[P_{sys}]$
SNM	NMC	100	9.1 M	6.5 min	7.35e-4	8.0e-5
	MC	115 k	1.9 B	9.7 day	—	8.0e-5
	UpB	—	175 k	1.3 min	8.70e-4	8.6e-5
RNM	NMC	100	9.9 M	7.5 min	1.92e-3	1.9e-4
	MC	58 k	0.95 B	5.4 day	—	1.9e-4
	UpB	—	105 k	2.5 min	2.18e-3	1.9e-4
WM	NMC	100	3.1 M	7.8 min	7.86e-5	8.1e-6
	MC	1.2 M	20 B	0.4 yr	—	8.1e-6
	UpB	—	60 k	1.1 min	8.98e-5	7.7e-6

By handling the global variations and the local variations at two different levels, the nested Monte Carlo method is able to utilize the benefits of the importance sampling technique for estimating the cell-level probability. The probability obtained is then analytically translated from cell-level to array-level at the global variation point. Compared with direct Monte Carlo method, the nested approach provides over 1000× speed-up in our experiments (both methods use RSM for sample evaluation).

Table 3 also lists the results of the upper bound (UpB) estimation. As expected, such method overestimates the failure probability by certain amount for all three examples. This is once more confirmed in Figure 4, where the WM failure probabilities for various array sizes are estimated with both the NMC and the UpB methods. For this 65nm process, however, the local variations dominant the overall effects. Therefore the overestimation by the UpB method is in fact only marginal. This suggests the UpB method as a candidate where runtime is more critical than accuracy (e.g. early-stage design decisions).

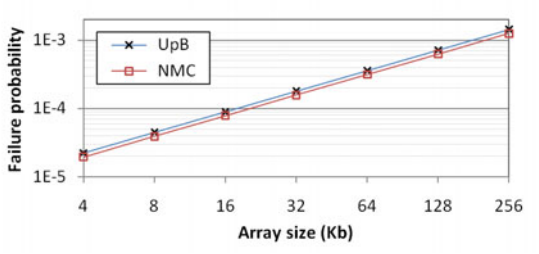


Figure 4. WM failure probabilities for different array sizes

6. CONCLUSIONS

With aggressive technology scaling, the design and analysis of SRAM circuits have become increasingly challenging. The most critical issues stem from the growing process uncertainties and the stringent functionality requirements. In view of the challenges, we propose a model-based importance sampling approach to facilitate the analysis of the rare failure events in SRAM circuits. Our proposed methodology extracts the SRAM failure probability at both the cell-level and the array-level. At cell-level, a piecewise modeling framework is applied to accurately model the cell stability metric over the large process variation space. A controlled sampling is then performed for better failure region coverage and sample reduction. At the array-level, we propose a nested Monte Carlo method that handles both the global and the local variations, as well as fully incorporates the benefits of the above cell-level failure probability extraction method.

In our experiments for a 65nm SRAM design, the piecewise model accurately captures the variation space up to ±6σ for the local variations. Our proposed methods for failure probability estimation also clearly outperform the conventional Monte Carlo approach at both the cell-level and the array-level. With the piecewise model and the fast probability extraction method, we accelerate the SRAM failure analysis by magnitudes and enable such difficult analyses.

As an extension to the proposed modeling and analysis flow, we are currently working to include design parameters (transistor sizes, array configuration parameters, etc.). The eventual goal is to help building up the complete methodology and tool-set to support the variation-aware design flow for SRAM systems.

7. ACKNOWLEDGEMENTS

The authors acknowledge the support of the Focus Center for Circuit & System Solutions (C2S2), one of five research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation program, and the National Science Foundation under contract CCF-0702278.

8. REFERENCES

[1] B. H. Calhoun, et al., "Digital circuit design challenges and opportunities in the era of nanoscale CMOS," *Proceedings of the IEEE*, vol. 96, no. 2, pp. 343-365, Feb. 2008.

[2] T. Mizuno, J. Okamura, and A. Toriumi, "Experimental study of threshold voltage fluctuation due to statistical variation of channel dopant number in MOSFET's," *IEEE Trans. on Electron Devices*, vol. 41, no. 11, pp. 2216-2221, Nov. 1994.

[3] A. J. Bhavanagarwala, X. Tang, and J. D. Meindl, "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," *IEEE JSSC*, vol. 36, no. 4, pp. 658-665, Apr. 2001.

[4] R. Heald, and P. Wang, "Variability in sub-100nm SRAM designs," *Proc. of ICCAD '04*, pp. 347-352, 2004.

[5] I. M. Sobol, *A Primer for the Monte Carlo Method.* CRC, Boca Raton, FL, 1994.

[6] 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. of DAC '06*, pp. 69-72, 2006.

[7] A. Singhee, and R. A. Rutenbar, "From finance to flip flops: a study of fast quasi-Monte Carlo methods from computational finance applied to statistical circuit analysis," *Proc. of ISQED '07*, pp. 685-692, 2007.

[8] R. L. Iman, J. C. Helton, and J. E. Campbell, "An approach to sensitivity analysis of computer models," *Journal of Quality Technology*, vol. 13, no. 3, pp. 174–183, Jul. 1981.

[9] C. Wann, et al., "SRAM cell design for stability methodology," *Proc. of IEEE VLSI-TSA '05*, pp. 21-22, 2005.

[10] K. Agarwal, and S. Nassif, "Statistical analysis of SRAM cell stability," *Proc. of DAC '06*, pp. 57-62, 2006.

[11] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS," *IEEE Trans. on CAD*, vol. 24, no. 12, pp. 1859-1880, Dec. 2005.

[12] A. Singhee, and R. A. Rutenbar, "Statistical blockade: a novel method for very fast Monte Carlo simulation of rare circuit events, and its application," *Proc. of DATE '07*, pp. 16-20, 2007.

[13] K. Takeda, et. al., "Redefinition of write margin for next-generation SRAM and write-margin monitoring circuit," *Proc. of ISSCC '06*, pp. 2602-2611, 2006.

[14] T. C. Hesterberg, "Advances in importance sampling," PhD Dissertation, Stanford University, Palo Alto, CA, 1988.

[15] J. Wang, X. Li, and L. T. Pileggi, "Parameterized macromodeling for analog system-level design exploration," *Proc. of DAC '07*, pp. 940-943, 2007.

[16] S. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Cambridge, United Kingdom, 2004.

[17] E. Seevinck, F. J. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *IEEE JSSC*, vol. 22, no. 5, pp. 748-754, Oct. 1987.

[18] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Prentice Hall, Englewood Cliffs, NJ, 2003.