

STATISTICAL PERFORMANCE MODELING OF SRAMS

A Thesis

by

CHANG ZHAO

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2009

Major Subject: Computer Engineering

STATISTICAL PERFORMANCE MODELING OF SRAMS

A Thesis

by

CHANG ZHAO

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Peng Li
Committee Members,	Gwan Choi
	Eun Jung Kim
Head of Department,	Costas N. Georghiadis

December 2009

Major Subject: Computer Engineering

ABSTRACT

Statistical Performance Modeling of SRAMs. (December 2009)

Chang Zhao, BS, Huazhong University of Science and Technology

Chair of Advisory Committee: Dr. Peng Li

Yield analysis is a critical step in memory designs considering a variety of performance constraints. Traditional circuit level Monte-Carlo simulations for yield estimation of Static Random Access Memory (SRAM) cells are quite time consuming due to their characteristic of low failure rate, while statistical methods of yield sensitivity analysis are meaningful for their high efficiency.

This thesis proposes a novel statistical model to conduct yield sensitivity predictions on SRAM cells at the simulation level, which excels regular circuit simulations in a significant runtime speedup. Based on the theory of the Kriging method that is widely used in geostatistics, we develop a series of statistical model building and updating strategies to obtain satisfactory accuracy and efficiency in SRAM yield sensitivity analysis.

Generally, this model applies to the yield and sensitivity evaluation with varying design parameters, under the constraints of most SRAM performance metrics. Moreover, it is potentially suitable for any designated distribution of the process variation regardless of the sampling method.

DEDICATION

To my parents and Xin.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the expert guidance of my advisor, Dr. Peng Li. I am very thankful for his encouragement and constant patience. From Dr. Li I have learned the scientific attitude toward academics and his dedication to research, which is and will be always helpful in my career life.

I would also like to thank my committee members, Dr. Gwan Choi and Dr. Eun Jung Kim, for their support and assistance throughout this research.

Many thanks to my friends and colleagues, who have provided many important materials and explanations of the manuscripts to help me get started in the research.

I also want to extend my gratitude to the department faculty and staff for making my graduate studies at Texas A&M University an exciting experience.

Finally, thanks to my mother and father for their unconditional love, support and belief in me. Thanks to Xin for his continuous encouragement and support.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
DEDICATION	iv
ACKNOWLEDGEMENTS	v
TABLE OF CONTENTS	vi
LIST OF FIGURES.....	viii
LIST OF TABLES	ix
CHAPTER	
I INTRODUCTION.....	1
A. Motivation	1
B. Previous Work and Our Contributions	3
C. Organization	4
II BACKGROUND.....	5
A. Standard 6T SRAM Cell and Data Operations.....	5
B. Stability and Performance Metrics	7
1. Static Noise Margin (SNM).....	7
2. Dynamic Noise Margin (DNM).....	8
3. Access Time.....	11
C. Design Variations	12
D. SRAM Design Objectives	13
III YIELD ESTIMATION USING KRIGING MODEL	14
A. Benefits of Kriging Method in SRAM Yield Estimation.....	14
B. Review of Kriging Model.....	16
C. Model Building Overview	18
D. Sample Technique and Updating Method	21
1. Sampling Technique.....	21
2. Updating Strategies	22

CHAPTER	Page
E. Model Stability and Convergence.....	25
F. The Algorithm.....	25
IV SENSITIVITY ANALYSIS	29
A. The Purpose of Sensitivity Analysis	29
B. Fast Yield Sensitivity Analysis for Multiple Design Parameters –Sub-Modeling Approach.....	30
V EXPERIMENTAL RESULTS	32
A. Yield Estimation under Constraint of Read/Write DNM	32
1. DNM Definition.....	32
2. Circuit Setup	33
3. Yield and Run Time Comparison	35
B. Yield Estimation under Constraint of Read/Write SNM.....	35
1. Circuit Setup	35
2. Yield and Run Time Comparison	36
C. Yield Estimation for Read Access Time	38
D. Boundary Convergence in Model Building Process	39
E. Yield Sensitivity Analysis and Model Verification.....	42
1. The Accuracy of Partial Kriging Model	42
2. Sensitivity Analysis	43
VI CONCLUSIONS AND FUTURE WORK	46
REFERENCES.....	47
VITA	51

LIST OF FIGURES

FIGURE		Page
1	SRAM Cell Level Optimization.....	2
2	Standard 6T SRAM Cell	5
3	SNM Definitions	8
4	Read DNM Definition.....	9
5	Write DNM Definition.....	10
6	Correlation between SNM and DNM in Read and Write	11
7	Model Building Process	20
8	Boundary Update.....	21
9	Sampling Method in Each Step.....	24
10	Circuit Setup for Read DNM and Write DNM	34
11	Circuit Setup for Read SNM and Write SNM.....	36
12	Circuit Setup for Read Access Time Measurement	38
13	6T SRAM Cell and Vth Assignments	39
14	The Boundary Figuring Process.....	40

LIST OF TABLES

TABLE		Page
I	Read/Write DNM Measurements in One Million Monte-Carlo Samples....	35
II	Read SNM Measurement in One Million Monte-Carlo Samples for Various Target Values.....	37
III	Write SNM Measurement in One Million Monte-Carlo Samples for Various Target Values.....	37
IV	Read Access Time Measurement in One Million Monte-Carlo Samples for Various Target Values.....	39
V	Accuracy of Partial Kriging Model.....	43
VI	Failure Rate Measurement and Yield Sensitivity Analysis for One Million Samples from Kriging Model and HSPICE.....	45

CHAPTER I

INTRODUCTION

A. Motivation

Static Random Access Memory (SRAM) cells are becoming increasingly important circuit components in most systems-level, very large scale integrated (VLSI) circuits such as microprocessors, while SRAM based memories (caches) impact area, power, timing, yield and schedule of the processor. As a result, SRAM is predicted to contribute to more than 90% of the die area in the future[1]. Therefore SRAM stability and effect on yield often determine the minimum voltage the chip can tolerate with acceptable yield. The direct relationship between voltage supply, yield, and SRAM stability highlight the need for robust and adaptive Cache design styles[2].

As IC device geometries continues shrinking, the transistor threshold voltages become harder to control in the sub-100nm design, which significantly decrease the stability margins in SRAM operations. Smaller SRAMs become more and more desirable with their popularity and domination on within-chip data retention in many applications. Meanwhile, the stability issue caused by process variations such as random dopant fluctuations in SRAM cell is an important concern in SRAM design, since single cell failures may introduce a dysfunctional memory unit. To tolerant single event upsets (SEU) induced soft errors, the dynamic and static noise margins in each operation

The journal model is *IEEE Transactions on Automatic Control*.

are served as the stability constraints in the yield analysis.

The optimization work of a 6T SRAM cell falls into two stages, one is to search for new optimal design points, while the other one is to evaluate the performance to verify the acceptability of the new point. In the cell level SRAM design optimization flow as shown in Figure 1, we have to iteratively perform the evaluation by analyzing the yield under different constraints (i.e. Noise Margin and Data Access Time) to determine applicability of the design point. The conventional SPICE simulation is time-consuming considering the state-of-the-art SRAMs' nature of low failure probabilities. Specifically, millions of simulations on a single SRAM cell need to be performed due to the dependences and quantities of their placement on a memory unit, which takes several weeks' time. Although SPICE simulation can be extremely accurate on every evaluation points, the binary fail/success decision nature of SRAM cell performance makes it only important to maintain extreme accuracy on the interested decision boundary.

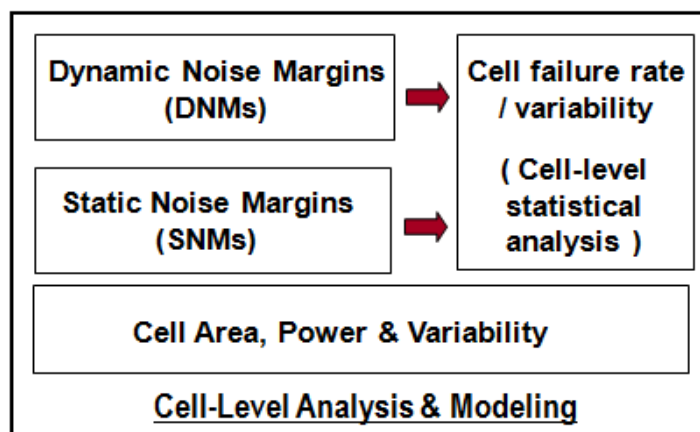


Fig.1. SRAM Cell Level Optimization

In summary, a special, fast and accurate method in yield analysis is desired in SRAM cell design.

B. Previous Work and Our Contributions

Current solutions for those problems are attempted by improving sampling techniques. The mixture importance sampling method proposed in [3] is based on the methodology of sampling to speedup Monte-Carlo simulations by efficiently capturing rare failure events. Similarly, the extreme value statistics theory and data filtering method in [4] used to enhance the speedup of Monte-Carlo simulation by SPICE like circuit level simulators, are also from the angle of sampling technique.

Our method is a statistical method focusing on improving the modeling of the simulation process instead of on sampling schemes as in [3] and [4]. It consists of a model building process through Kriging method and an updating process to refine it. High efficiency can be achieved by utilizing so called “zoom in” effect of Kriging modeling. Additionally, cooperation with aforementioned sampling techniques is possible to obtain further profound speedups.

Besides yield analysis for fixed design parameters, we can also apply the concept and procedures to sensitivity analysis. Using our method, we can efficiently get the yield sensitivity of any particular design parameter, by which, the yield difference can be easily prospected and reasonably estimated if we change that design parameter within 10%.

By integrating our method into the design optimization flow, one can accurately and quickly get the performance evaluation under various performance metrics, proceeding to determining whether or not keep the design and where to move to search for the next optimal point.

C. Organization

This thesis is organized as follows. In Chapter II, we introduce some crucial preliminaries in SRAM performance metric and design flow. Then the Kriging method is proposed to speedup yield analysis in 6T SRAM cell, which is detailed in theory and practical execution in Chapter III. In Chapter IV, we introduce the theory and methodology of yield sensitivity analysis respective to design parameters. Experimental results and an example of the integration of our method with SRAM optimization are presented in Chapter V. Finally, we arrive at the conclusions and future works in Chapter VI.

CHAPTER II

BACKGROUND

A. Standard 6T SRAM Cell and Data Operations

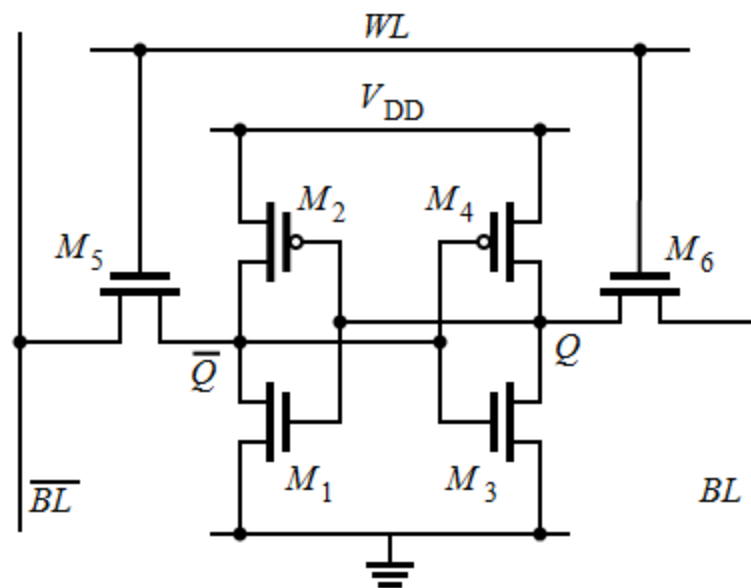


Fig.2. Standard 6T SRAM Cell

Each bit in an SRAM is stored on four transistors that form two cross-coupled inverters. This storage cell has two stable states which are used to denote logical “0” and logical “1”. M_2 and M_4 are “pull up” transistors, which draw current down from the supply voltage (V_{DD}) to denote a 0, while M_1 and M_3 are “pull down” transistors which draw current to the ground to denote a “1”. Two additional access transistors serve to control

the access to a storage cell during read and write operations. Access to the cell is enabled by the word line (WL in Figure 2) that controls the two access transistors M_5 and M_6 which, in turn, control whether the cell should be connected to the bit lines: BL and \overline{BL} . They are used to transfer data for both read and write operations. The strengths of the transistors determine the performance and stability of the cell.

There are three basic statuses for SRAM: hold, read and write.

Hold: When the circuit is idle, the word line is not asserted, the access transistors M_5 and M_6 disconnect the cell from the bit lines. The two cross coupled inverters formed by $M_1 - M_4$ will continue to reinforce each other to be held at the original storage status as long as they are connected to the supply.

Read: When data has been requested, bit lines (BL and \overline{BL}) are both pre-charged to “1” at the beginning of the read cycle. Then WL is asserted, enabling both access transistors. After that, the values stored at Q and \overline{Q} are transferred through M_6 and M_5 to BL and \overline{BL} respectively. If a bit “1” is stored at Q (“0” at \overline{Q}), BL will be pulled up toward VDD by M_4 and M_6 and \overline{BL} will be discharged through M_1 and M_5 to “0”. While the content of the memory is a “0”, the opposite would happen, pulling BL down to “0” and \overline{BL} up to “1”.

Write: When the content of the memory needs to be updated, a write operation occurs. The start of a write cycle begins by applying the value to be written to the bit lines, i.e. setting \overline{BL} to “1” and BL to “0” if we wish to write a “0”. A “1” is written by inverting the values of the bit lines. WL is then asserted and the value to be stored is latched in. The reason of this works is that the bit line input-drivers are designed to be

stronger than the relatively weak transistors in the cell itself, so that they can easily override the previous state of the cross-coupled inverters. Careful sizing of the transistors in an SRAM cell is necessary to ensure proper operation.

B. Stability and Performance Metrics

There are several metrics used to evaluate the SRAM stability and performance, including Static Noise Margin (SNM), Dynamic Noise margin (DNM), and data access time.

In general, Noise Margin is the maximum spurious signal that can be accepted by the device when used in a system while still maintaining the correct operation. If the consequences of the noise applied to a circuit node are not latched, such noise will not affect the correct operation of the system and thus be deemed tolerable[5]. The noise margins in hold manifest the capabilities of data retentions, while in read and write, they reflect the robustness of nondestructive reads and efficient writes under single event upsets (SEU).

1. Static Noise Margin (SNM)

SNM has been discussed a lot in [1, 6, 7], serving as one of the performance metrics for SRAM static stability. Figure 3 shows the circuit diagram of an SRAM cell and the definition of the two most important noise margins involved in its operation.

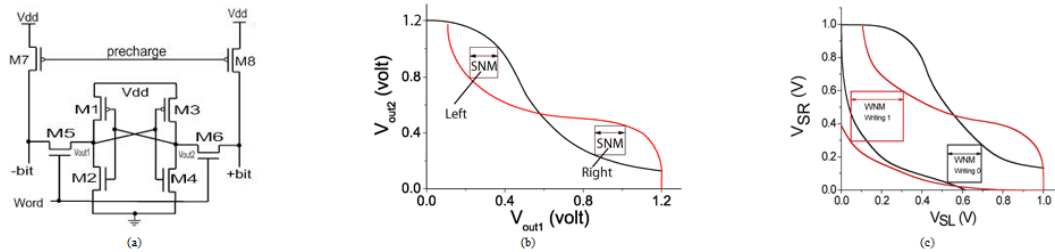


Fig.3. SNM Definitions

The Static Noise Margin is defined to be the DC noise voltage required to obtain the opposite value to the existent information stored in the cell during a read period, as shown in Figure 3 (b), and the Write Noise Margin is the DC noise voltage needed to result in a failure of completing the change of a cell status as intended during a write period, which is shown in Figure 3 (c)[8].

2. Dynamic Noise Margin (DNM)

SNM assumes the noise to be “static” or DC, indicating the noise will last for an infinite duration, while DNM employs a much more realistic picture, which takes the duration, amplitude and charge of the injected noise all into consideration. In the dynamic read and write operation, the timing of wordline plays a critical role in enhancing their noise margins. As to improve dynamic stability, DNM is strongly desirable in the SRAM analysis in that it provides precise timing control.

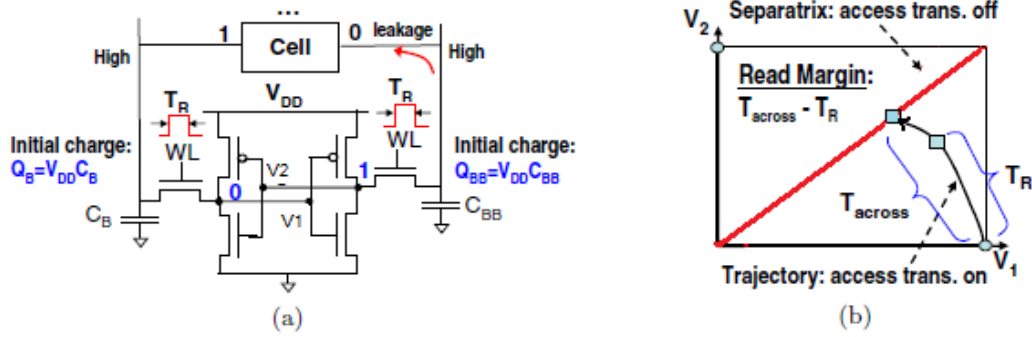


Fig.4. Read DNM Definition

At the end of the read operation, the cell returns to hold after the wordline goes off. However, before that occurs, the cell could be pushed away from the initial state because of noise during the read process. This indicates a failed read operation since it flips the memory state. The separatrix is proposed to be the discrimination standard of whether the state is flipped or not in [9]. As described, the separatrix, i. e. the stability boundary, is a 45 degree line for symmetric SRAM cells. That is to say, we perform transient simulations and sweep the values of V_1 and V_2 , as in Figure 4 (a), until they get equal to each other, then we record the time as T_{across} . The read DNM is defined as

$$T_{DNM,R} = T_{across} - T_R \quad (2.1)$$

where T_R is the predetermined read time, as shown in Figure 4 (b). Compared to read DNM, read SNM produces a pessimistic estimation. Since the read time is impossible to be infinite as in SNM, even if the SNM predicts a state flip, in reality, the read time is not long enough to make it happen.

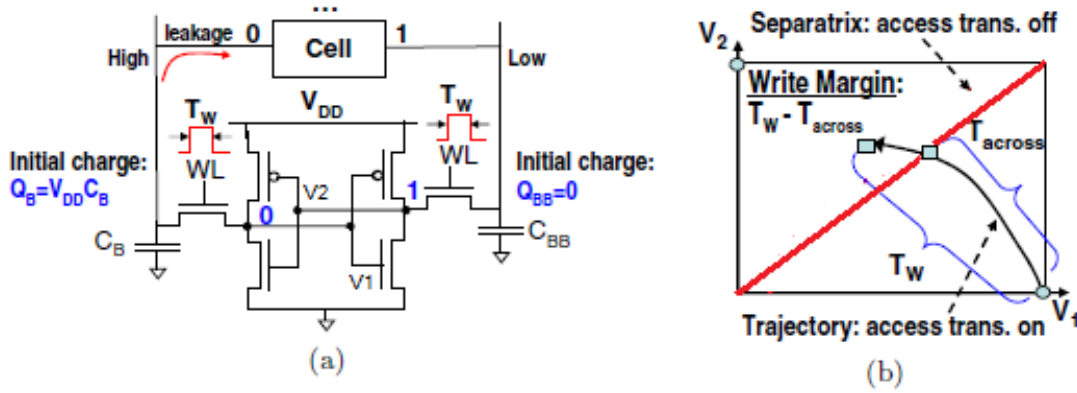


Fig.5. Write DNM Definition

Similarly, in write operation shown in Figure 5, the write DNM is defined as

$$T_{DNM,W} = T_W - T_{across} \quad (2.2)$$

where T_W is the predetermined write time controlled by the wordline pulse width. While the write time is also impossible to be infinite as assumed by the SNM, write SNM provide an optimistic estimation for the dynamic property. In fact, even if the SNM predicts a successful write operation, the write time may be not long enough for the new data to overwrite the old one.

To highlight the importance of dynamic stability, we have examined the correlations between SNMs and DNMs in read and write operations. Totally 10000 random samples of a 6-T SRAM cell are experimented on with the uniformly distributed process variations to reach every corner of the variation space. It is assumed that the read and write time are both 10ps. To clarify the details of the performances, we have filtered out the extreme values that are far away from the main cluster to zoom-in. After normalizing

the SNM and DNMs, the results are shown in Figure 6, demonstrating that no strong correlation between SNM and DNM of read and write operations exists.

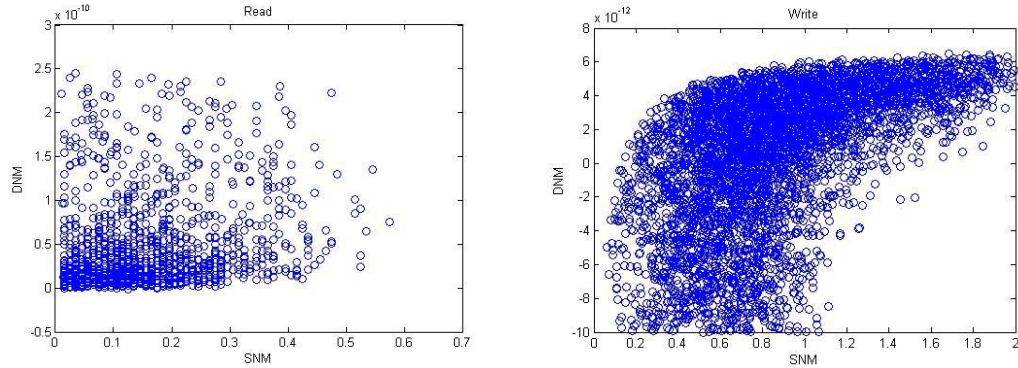


Fig.6. Correlation between SNM and DNM in Read and Write

Therefore, we are unable to predict the tendencies of DNMs from the corresponding SNMs. It is meaningful to explore the DNMs as well to provide design guidance for SRAM.

3. Access Time

Access time is the time a SRAM cell locates a single piece of bit information either from or to the bit line in write or read operation. Ideally, the access time of memory should be fast enough to keep up with the Central Process Unit (CPU) to maintain the number clock cycles. In other words, access time is an important metric to evaluate the speed of the memory accesses.

In this thesis, we take five metrics as examples, which are Read SNM, Write SNM, Read DNM, Write DNM and Read Access time. The reason we choose these typical metrics is that, read and write operations are more complicated than hold status,

and the methodology can be easily applied to hold stabilities. For the measurement of access time in write, it can be treated the same as Write DNM.

The concept of SNM we employ is from[1, 6, 7], and DNM is from[9].

C. Design Variations

Variation roughly consists of a random component and a global component. The random component differs from transistor to transistor, such as Random Dopant Fluctuation (RDF). The global component varies from one diffusion process to another and from one slice to another, such as transistor oxide thickness. The random component occurs quite irregularly and is thus very difficult to control. On the other hand, the global component shows the same tendency to the entire chip[6].

MOS threshold voltage variation due to random fluctuations in the number and location of dopant atoms is an increasingly significant effect. RDF is assumed to be the major contributor to device mismatch of identical adjacent devices. Along with the degradation of devices, they form a big challenge for designers to build circuits that both yield high under the influence of process variations and remain functional with respect to long term device drift.

To model the process variations, we randomly sample the six threshold voltage values (V_{th}) independently around the nominal value in the model list to represent the random component. For the global component, we use the same oxide thicknesses (T_{ox}) for the six transistors in one particular SRAM cell, but varied values for different cells.

D. SRAM Design Objectives

Circuit design is relatively more important since the logic and function for a SRAM cell is ready to use. As a series of performance targets is previously set, we perform yield analysis toward various metrics to see whether the performance targets are met after each circuit design cycle. If not, we go back to change the design parameter so that the aims can be satisfied with a decent yield level. The sensitivity analysis can contribute to providing the direction and magnitude of the next move in optimizing design parameters.

CHAPTER III

YIELD ESTIMATION USING KRIGING MODEL

A. Benefits of Kriging Method in SRAM Yield Estimation

The traditional method of SRAM yield estimation for a fixed design is to perform millions of SPICE simulations on SRAM cells whose process variations are Gaussian distributed. This is unaffordable for the sake of efficiency. Alternatively, we employ Kriging method, which is an unbiased estimation procedure using known values to determine unknown values[10]. We treat each SRAM cell with process variation as a data point with certain design parameters in a multi-dimensional space. The performance distribution of the SRAM cell points to be evaluated is estimated from the training data points whose performances are known by SPICE simulations. Kriging generates an optimal set of weighting factors from the spatial continuity information of the training data points, and use them to interpolate the performance value with process variations for a certain SRAM cell from the nearby vectors, where closer points are weighted more heavily than distant ones in the calculation. In a word, from the training data points, the Kriging method statistically analyzes the variation in performance values over different distances to generate an optimal set of weighting factors, based on which the performances of the evaluating points are predicted and the estimation errors are minimized.

As described in [11], the features of Kriging Model are listed as follows.

First, Kriging Model is able to capture strong nonlinearity in circuit performance depending on process or designable parameters in global parameter space.

Second, for a specified modeling accuracy, the number of sampling points measurement needed by Kriging Model is much smaller than the traditional method. So the runtime of circuit-level simulations can be largely reduced.

Third, Kriging Model is convergent for extrapolation, i.e. its extrapolation goes to the average value of sample points instead of infinity as traditional Response Surface Model (e.g. quadratic polynomial) does. This property is desirable for global approximation of circuit performance.

Therefore, by applying Kriging method, we use a small number of selected SPICE simulations to “map” the performance response surfaces of interest as functions of the controllable process or decision variables[12]. In SRAM yield analysis, high precision is needed in predicting the performance that is close to the critical fail/success discrimination value to conduct an accurate prediction, while the description of the cell whose performance is far away from the critical value just needs a rough estimation as long as it does not cross the critical value. Ideally, the more training data points are located in certain area, the more accuracy can be obtained in this region, which is called the “zoom-in” effect. Taking advantage of it, we can achieve high efficiency by deliberately changing the distribution of training data points, so as to place more weights on critical area while ignore some details of unimportant area.

B. Review of Kriging Model

The Kriging model originated from study in geo-statistics which is used to obtain the response surface from a set of spatially correlated data. While computer simulation is inherently in absence of random errors, the approximation with uncertainty is realized by the stochastic modeling approach using Kriging model. The circuit performance $Y(x)$, which is treated as a realization of a random field due to process variations, can be expressed as a linear combination of a regression model and a stochastic process[11]:

$$Y(x) = \beta \mathbf{f}^T(x) + Z(x) \quad (3.1)$$

where x is a k –dimensional vector standing for k process variations; the term $\mathbf{f}(x)\beta^T$ is the regression portion globally representing the response surface, in which $\mathbf{f}(x) = [f_1(x) f_2(x) \dots f_k(x)]^T$ is a vector of provided regression functions, and $\beta = [\beta_1 \beta_2 \dots \beta_k]^T$ is a vector of unknown regression coefficients; $Z(x)$, with a zero mean and a process variance σ^2 , is the stochastic process that captures the localized deviations from the global regression portion, in order for Kriging model to make interpolation among sample points. The correlation function of $Z(a)$ and $Z(b)$ is $R(a, b)$, a product of stationary one-dimensional correlations and modeled by the power exponential correlation family[13] that is widely used in computer experiments literature.

$$R(a, b) = \prod_{i=1}^k \exp(-\theta_i |a_i - b_i|^{p_i}) \quad (3.2)$$

where θ and p are both k -dimensional vectors of unknown coefficients, with $\theta_i \geq 0$ and $0 \leq p_i \leq 2$. We set $p_i = 2$ for all ' p 's to make $R(a, b)$ infinitively differentiable at zero[14]. Therefore, the covariance matrix of $Z(x)$ is:

$$\text{Cov}(Z(a), Z(b)) = \sigma^2 R(a, b) \quad (3.3)$$

Equation (3.2) is related to the “distance” between two points. While the distance between two points are further, $Y(a)$ and $Y(b)$ are less likely to stay at the same value. The decreasing rate of the likelihood is higher for a larger θ_i value. A larger p_i renders Kriging model greater smoothness[11].

To model the performance evaluation from equation (3.1), we need to estimate unknown parameters θ , β and σ^2 by performing n sets of simulations on n sets of input process variation vectors $S = [s_1, \dots, s_n]^T$, where $s_i = [x_{1,i}, \dots, x_{k,i}]$. Suppose the corresponding performance simulation result is $Y_S = [Y_{s_1}, \dots, Y_{s_n}]^T$ and R is a $n \times n$ matrix of correlations, where $R_{i,j} = R(s_i, s_j)$. The performances of the new point x_0 to be evaluated are denoted by $r(x_0) = [R(x_0, s_1), \dots, R(x_0, s_n)]^T$ and $F(x) = [\mathbf{f}(s_1), \dots, \mathbf{f}(s_n)]^T$. Then the predictor is derived as[15]:

$$\hat{Y}(x_0) = \hat{\beta} \mathbf{f}^T(x_0) + r^T(x_0) R^{-1} (Y_S - \hat{\beta} F) \quad (3.4)$$

where $\hat{\beta}$ is estimated by using least squares regression:

$$\hat{\beta} = (F^T R^{-1} F)^{-1} (F^T R^{-1} Y_S) \quad (3.5)$$

Given[13, 16]:

$$\hat{\sigma}^2 = (Y_S - \hat{\beta}F)^T R^{-1} (Y_S - \hat{\beta}F) / n \quad (3.6)$$

The mean square error (MSE) for the performance predictor can be obtained from:

$$MSE[\hat{Y}_0] = \sigma^2 \left(1 - [\mathbf{f}^T \quad r] \begin{bmatrix} 0 & F^T \\ F & R \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f}(x_0) \\ r(x_0) \end{bmatrix} \right) \quad (3.7)$$

In addition, the prediction accuracy is evaluated by the relative prediction uncertainty[17]:

$$E = \sqrt{(MSE[\hat{Y}_0]) / \hat{Y}_0} \quad (3.8)$$

More details about the implementation of Kriging model can be found in [18].

C. Model Building Overview

We use a seven-dimensional space to model the process variations for a 6T SRAM cell, of which six are the threshold voltages(Vth) for six transistors respectively and one is the oxide thickness shared by all of them in a single cell. If we predetermine a critical value for the performance metric, a boundary surface will be formed to divide the successful and failed region. In other words, cells will cover the whole space, and we use their performances to evaluate them. If its performance is better than the critical one, it falls to the successful side towards the boundary, while worse performance goes to the failure side. The model is supposed to provide moderate accuracy over the region which is evenly covered by the input training data space. As a matter of fact, failures for SRAM

cells are rare events, therefore, many sampled cells will be far away from the boundary, we call them as the “safe points”, whose values are not necessarily to be very precise to obtain a right determination of failure or success. In the contrast, the so called “dangerous points” are cells with the performances that are very close to the critical value.

To conduct efficient and accurate analysis, we are dedicated to revising the training data with less simulation but better configuration of the boundary. According to the nature of binary decision, we have to get precise performance evaluation on the points that are adjacent to the boundary since a small deviation might cause an opposite decision, a wrong decision possibly. Considering the zooming-in attribution of Kriging model, the more simulations around the boundary region are run, the better representation of the boundary surface is obtained. Thus we iteratively enroll more such “boundary points” into the training data set to update the model until its convergence. Alternatively, another strategy is to add new points with high uncertainty values into training data to improve the accuracy.

The flow chart of the model building process is as following in Figure 7.

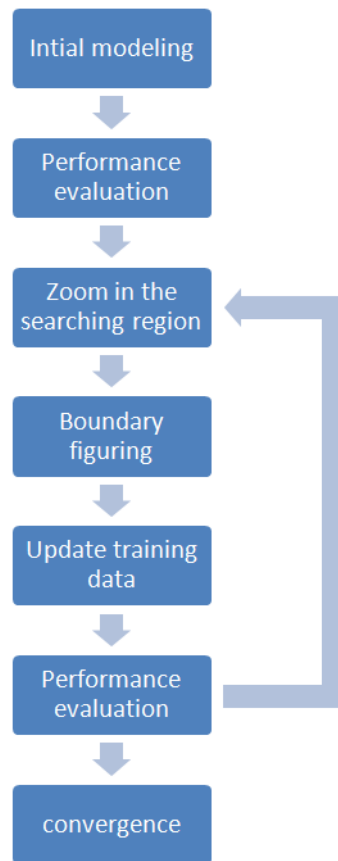


Fig.7. Model Building Process

Each iteration in this flow updates the Kriging Model in the way that is shown as following in Figure 8:

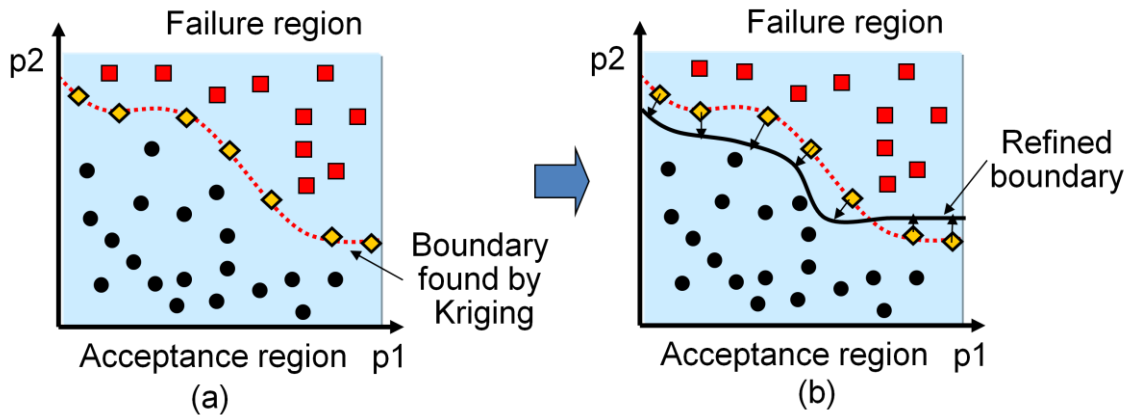


Fig.8. Boundary Update

D. Sample Technique and Updating Method

To optimize the model performance, we need to balance its speed and accuracy. Theoretically, the more simulations on training data are done, the more accurate the results are yield, and the longer runtime it takes. Therefore, we studied a series of strategy of selecting proper training data to deal with this trade-off.

1. Sampling Technique

The final target is that we use the build-up model to predict the performance of a given sample space to calculate the yield. In the following paragraphs, the “space” indicates the final evaluation sample space. For example, as for yield analysis, the final evaluation space is always a Gaussian distribution with a certain sigma and mean value.

Initially, the model is supposed to get a rough description over the whole space that is going to be evaluated. So in the first set of training data, it should be able to

capture the characteristic of the entire space. We build the initial model by using a 70-simulation Latin Hypercube Sampling technique, which makes the input parameter combinations spread evenly in the parameter space[19]. The sample range is $\pm 5\sigma$ away towards the mean, so that it covers any rare event corner. Details of Latin Hypercube Sampling can be found in[20].

The evaluation sample we use in the model building process is not the final Gaussian distributed one. The reason is that, if used, only a small amount of failures will be presented due to the low failure rate nature of SRAM cells. In that circumstance, we cannot efficiently figure out the boundary even by millions of evaluations. Consequently, for the initial performance evaluation, we use the same space filling sampling as in the training data, but enlarge the quantity to 100000, large enough to uniformly cover the entire variation space.

2. Updating Strategies

The sequential step is to enroll more particular training data to figure out the decision boundary meanwhile keep the first set of uniform training data to maintain the overall perspective.

After the initial model is built up and the performance evaluation is done by it, we can roughly figure out a boundary through the prediction in the first iteration. We choose points that are very close to the predicted boundary with tolerable error and points with high error predictions to do more simulations, comparing the result to the target performance values to see whether they are really around the boundary. We keep the points that are simulated to be on or around the boundary into the next iteration.

Meanwhile, through the first evaluation, we discard those samples, whose calculated performance metrics are further than $\pm 30\%$ away from the target performance as we are confident that the initial model can make a right decision for them. In the subsequent model building steps, we use these sifted samples to do the experiments.

The sample and updating strategy are summarized as following in Figure 9:

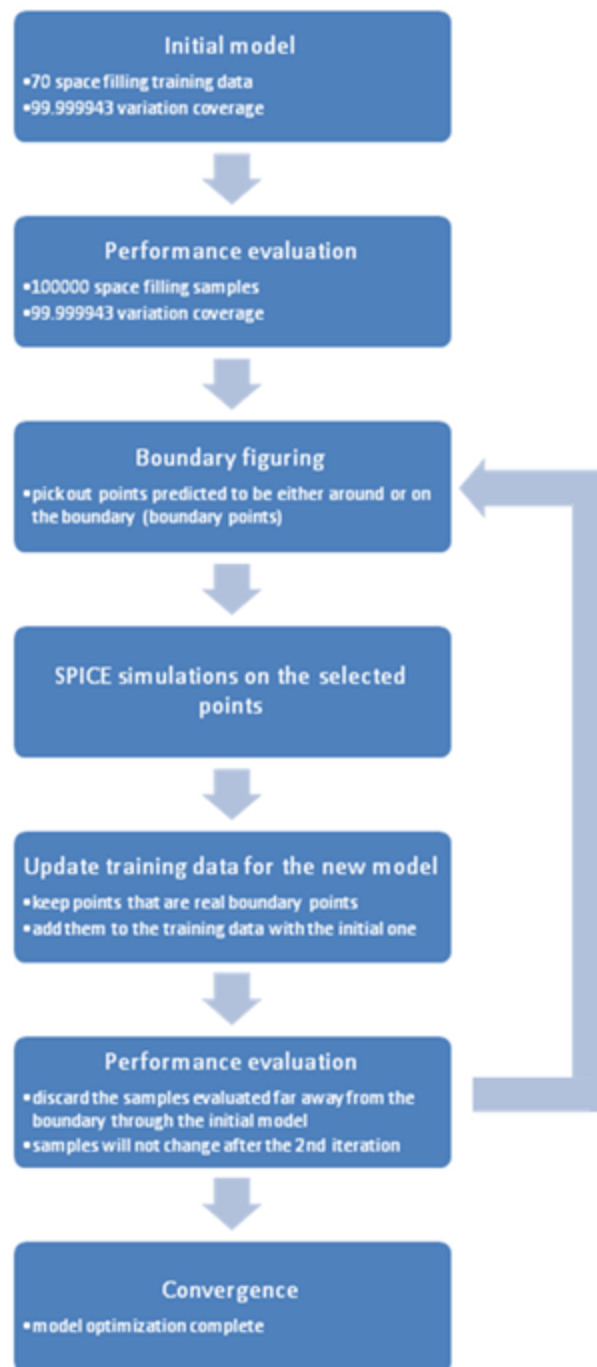


Fig.9. Sampling Method in Each Step

E. Model Stability and Convergence

The model building process will suffer the matrix singularity if no control is taken when selecting the training data. Furthermore, the cause of singularity is the precision limitation of MATLAB. To be specific, when two points in the training data are too close to each other, they will be identified as the same point by the program. In that case, mathematically, the two corresponding lines in the matrix express are regarded no different, leading the matrix to be singular, which stops the numerical calculation as a result. Considering the singularity problem may disturb the stability of the model and reduce the accuracy, we manage to perform a “Distance check” to avoid it.

The distance check process is realized as following. Before SPICE simulations are performed, on every newly added training data point, we calculate its distances to all the old ones from the previous iterations. If one of the distances is too small, we discard this point and proceed to the next one. If all the distances are big enough, we accept this point and add it to the training data. Then while it comes to the next new point, it will as well calculate its distance to all the present points, including the last added one.

This distance check procedure is also applied to the determination of model convergence. If there is no new selected training data points survived after the distance check, which means the old training data is already capable of defining a precise boundary, the boundary is converged and the final model is built up.

F. The Algorithm

Algorithm: Yield calculation through Kriging model

Input: Circuit netlist, target performance T , Process variation test samples $S = [s_1, \dots, s_N]$, number of test samples N , maximum error tolerance E_{max} , minimum tolerant distance D_{max} between two points.

Output: The yield $Y=Y(S,T)$.

- 1: Evenly sample the variation space using m points H_1, \dots, H_m and simulate the corresponding performances P_1, \dots, P_m through SPICE.
- 2: Construct the initial Kriging model via the training data set $\{H_i, P_i\}$ from step 1. Evenly sample the variation space using $n=N$ points V_1, \dots, V_n . Evaluate the performance t_1, \dots, t_n and the MSE e_1, \dots, e_n of them by the initial Kriging model.
- 3: for $i = 1$ to n do
 - 4: compare t_i with target performance T , and e_i to E_{max} ,
 - 5: if $t_i < 70\% * T$ or $t_i > 130\% * T$ then
 - 6: delete $\{V_i, t_i\}$ from the sample points to build the model, $n=n-1$
 - 7: end if
- 8: end for
- 9: for $i = 1$ to n do
 - 10: compare t_i to target performance T , and e_i to E_{max} , reset the counter, $c=0$
 - 11: if $99.99\% * T < t_i < 100.01\% * T$ or $e_i > E_{max}$ then
 - 12: keep V_i into the potential new training data h_1, \dots, h_c , $c=c+1$
 - 13: end else if
- 14: end for
- 15: for $i = 1$ to c do

```

16: calculate the distance  $d_{ij}$  from  $h_i$  to  $H_j \in [H_1, \dots, H_m]$  one by one
17: calculate the distance  $d_{ik}$  from  $h_i$  to  $h_k \in [h_{i+1}, \dots, h_c]$  one by one
18: if for any distance, there exist,  $d_{ij} < D$  or  $d_{ik} < D$ 
19:     delete  $h_i$  from the potential training data  $c=c-1$ 
20: end if
21: end for
22: if  $c=0$  then,
23:     finish the iterations, keep the final set of training data as to build the ultimate
        Kriging model, go to step 35
24: end if
25: else if  $c>0$  then
26: do SPICE simulation for  $h_1, \dots, h_c$  to get the performances  $p_1, \dots, p_c$ 
27: for  $i=1$  to  $c$ 
28:     compare  $p_i$  with target performance  $T$ ,
29:     if  $99.99\%*T < p_i < 100.01\%*T$  or  $e_i > Emax$  then
30:         include  $\{h_i, p_i\}$  into training data  $\{H, P\}, m=m+1$ 
31:     end if
32: end for
33: update the Kriging model, and evaluate the performance of  $V_1, \dots, V_n$  using the
        updated model, repeat the process from 9 to 32, until  $c=0$ .
34: end if

```

35: use the final optimized Kriging model to calculate the performance $F = [f_1, \dots, f_N]$, of the given test samples $S = [s_1, \dots, s_N]$, and compare the results with the performance target value to get the yield $Y=Y(S,T)$

CHAPTER IV

SENSITIVITY ANALYSIS

A. The Purpose of Sensitivity Analysis

Due to the variation in the failure probability, the choice of the transistor design parameters has a strong impact on the yield. Hence, it can be concluded that a statistical approach to the design of transistor is possible, and necessary, to maximize the yield. The sensitivity analysis model can be effectively used for such statistical optimizations.

From the sensitivity information, we can easily anticipate the impact of the design parameter adjustment in each particular transistor on the yield for diverse performance metrics. This is important for SRAM designers to improve the yield in their optimization.

In the memory design cycle, after the yield is given by the model for a set of fixed design parameters, we need to determine the next move for adjusting them for a better yield in the optimization process. The searching decision includes both the direction and the magnitude of the next pace. Intuitively, we can anticipate the direction of the yield change from the previous design toward circuit analysis. However, the magnitude of the impact for each design parameter cannot be precisely scaled in that manner. To deal with this problem, we propose the sensitivity analysis to describe the statistical changes in yield after tuning related design parameters within a 10% range for the optimization stage that follows.

While doing the yield analysis for a given design with parameters $D=[D_1, \dots, D_n]$, we also provide the yield prediction for the design varying within 10% by perturbing of one of the design parameters. In other words, when D_i change by 10%, the correspondingly varied yield of the design can be assessed by sensitivity analysis via our model. From this information, we can reasonably determine the next searching point of this design parameter in the optimization. Then sensitivity is calculated by:

$$S_i = \frac{\partial Y}{\partial D_i} = \frac{Y_i - Y_0}{\Delta D_i} \quad (4.1)$$

where Y_0 is the yield estimation of the nominal design, Y_i is the yield estimation of changing the i th design parameter within 10% and ΔD_i is the magnitude of the changing. We assume that within this small range, the difference of yield is linearly added up by the influence of each design parameter, that is:

$$Y = Y_0 + S_1 \Delta D_1 + \dots + S_n \Delta D_n \quad (4.2)$$

However, it should be noted that this is not applicable when the difference of the design parameter is larger than 10% since the linear approximation is valid in a small range.

B. Fast Yield Sensitivity Analysis for Multiple Design Parameters–Sub-Modeling Approach

Intuitively and reasonably, while we perturb one of the design parameter within 10%, the boundary points in the process variation space would not change much. Therefore, after the nominal yield value for the fixed design is calculated, we can use sub-modeling for sensitivity analysis, which means the direct utilization of the training data input selection

(in the last stage) on the yield analysis for a neighboring design point. This strategy saves us the time of the iterations in rebuilding the model. The only thing we need to do is to run simulations on the perturbed design using the selected points we have, and to build the model to calculate the yield once for each design parameter. Instead of running simulation for the perturbed design, additional speedup can be gained by calculating the results through the sensitivity information from the previous simulations on the selected training data in the nominal design.

As the yield estimations are made by the sub-modeling approach, we calculate the sensitivity of each design parameter through equation (4.1).

CHAPTER V

EXPERIMENTAL RESULTS

For fixed nominal design points, under the constraints of different performance metric, yield comparison between standard Monte-Carlo simulation using HSPICE and Kriging model is made, and to be presented. We also experiment on partial-Kriging model building process, which employ the training data directly from a 5% biased design as described in the sensitivity analysis to show the accuracy and advantage of sensitivity analysis methodology.

The future size of the transistors denoted as FS is 45nm. The failure rates are all measured from one million Gaussian distributed samples.

A. Yield Estimation under Constraint of Read/Write DNM

1. DNM Definition

The definition of DNM comes from [9], we use a 45 degree line as the referenced separatrix for approximation.

The circuit is shown in Figure 9. In read, we perform transient simulations examining the voltages of Q and \bar{Q} in sufficient time duration. If the two values happen to be the same, which indicates the state-flip, we stop and record the time used. The time value and the pre-determined target is compared to see whether the cell is qualified.

In write, similar to read, we measure the time needed for the state-flip. The only difference is that, for a read operation, a state-flip informs a failure, whose time is the

longer the better for a cell. However, in write, a short-time state-flip implies a fast writing success in this case. Therefore, write DNM can be also treated as write access time.

The yield value is calculated as following:

For read:

$$f(x_i) = \begin{cases} 0, & (t(x_i) < Tr; x_i \in R^n) & \text{fail} \\ 1, & (t(x_i) \geq Tr; x_i \in R^n) & \text{success} \end{cases} \quad (5.1)$$

For write:

$$f(x_i) = \begin{cases} 0, & (t(x_i) \geq Tw; x_i \in R^n) & \text{fail} \\ 1, & (t(x_i) < Tw; x_i \in R^n) & \text{success} \end{cases} \quad (5.2)$$

where Tr and Tw are the target values of read and write DNM, R^n is the 7 dimensional variation space, $t(x)$ is the state-flip time either measured from HSPICE simulation or calculated from our model. Then we have

$$\text{Yield} = \frac{1}{N} \sum_{i=1}^N f(x_i) \quad (5.3)$$

in which N is the number of SRAM cell samples.

2. Circuit Setup

The circuit setup and equivalent circuit for analysis are shown as below in Figure 10:

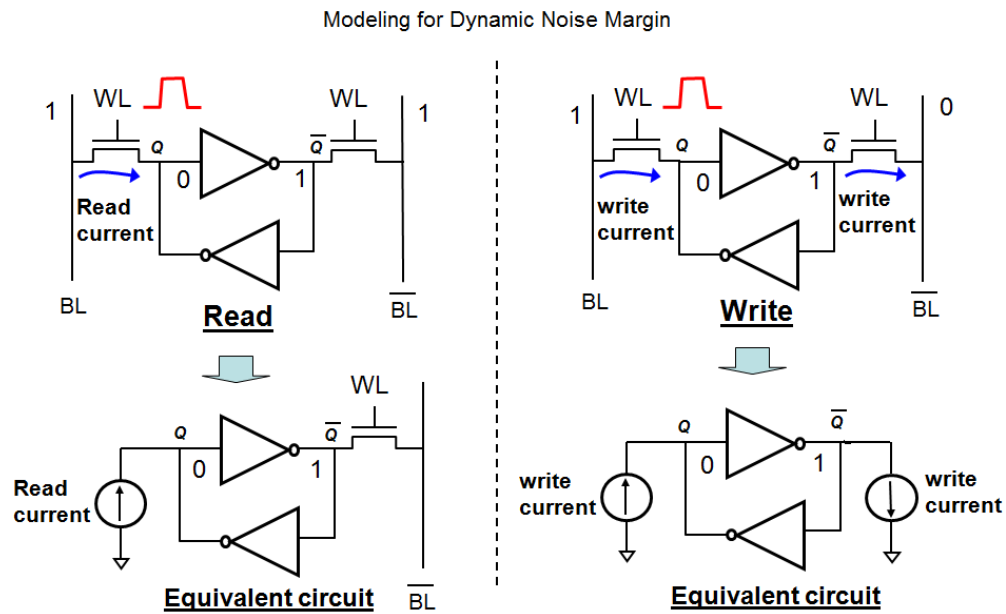


Fig.10. Circuit Setup for Read DNM and Write DNM

We consider three design parameters in the symmetric cell, including the width of NMOS pull-down transistors(W_n), the width of PMOS pull-up transistors(W_p) and the width of NMOS access transistors(W_{ax}). We denote $W=[W_n \ W_p \ W_{ax}]$ as the design parameter matrix. The value of this matrix is set to be different to avoid 100% yield, which is not suitable for the demonstration of the model.

$W = [4 \ 2.5 \ 1.875] * FS$ in Read DNM and $W = [4 \ 2.5 \ 7] * FS$ in Write DNM.

3. Yield and Run Time Comparison

Yield and run time comparison among SPICE Monte-Carlo simulation, full model building simulation and partial-model building simulations is presented in the following Table I.

Table I. Read/Write DNM Measurements in One Million Monte-Carlo Samples

	HSPICE	Kriging Model
Failures for read ($T_r=1\text{ns}$)	134	143
Failures for write($T_w=10\text{ps}$)	832	871
Run time	300hours	5hours

B. Yield Estimation under Constraint of Read/Write SNM

1. Circuit Setup

The read and write SNM measurement has been discussed a lot in [1, 6, 7], for which DC analysis is performed. We sweep the V_n from 0.1V to 2V, and voltages of Q and \bar{Q} are measured in each DC sweep to determine at which value of V_n , the cell starts to behave incorrectly. The yield can be calculated by equation (5.1), (5.2), (5.3) by replacing the time values by the voltage values.

The circuit we used to perform the measurement through HSPICE simulation is as following in Figure 11:

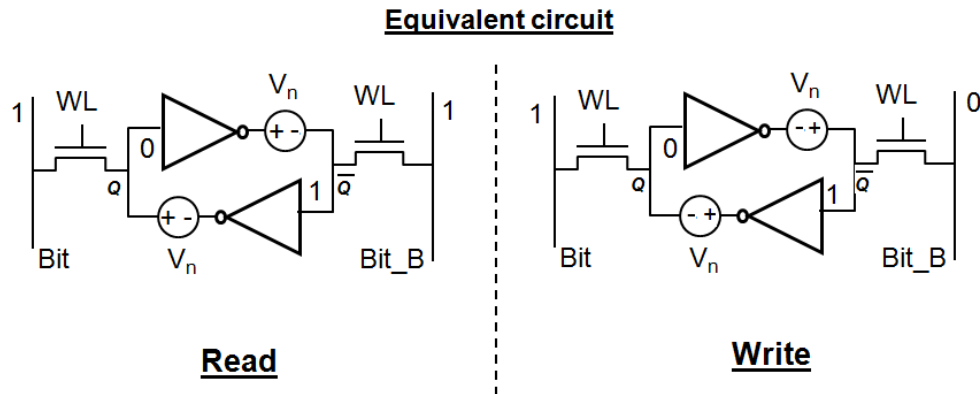


Fig.11. Circuit Setup for Read SNM and Write SNM

$W = [4 \quad 2.5 \quad 1.875] * FS$ in Read SNM circuit and $W = [4 \quad 2.5 \quad 5] * FS$ in Write SNM.

2. Yield and Run Time Comparison

Yield and run time comparison between SPICE Monte-Carlo simulation and full model building simulations is presented in the Table II and Table III, read and write respectively.

Table II. Read SNM Measurements in One Million Monte-Carlo Samples for Various Target Values

Critical voltage value	Failures from HSPICE	Failures from full Kriging model
0.18	11	11
0.24	149	158
0.34	4443	4418

Table III. Write SNM Measurements in One Million Monte-Carlo Samples for Various Target Values

Critical voltage value	Failures from HSPICE	Failures from full Kriging model
0.17	11	17
0.2	45	67
0.3	2180	2017

C. Yield Estimation for Read Access Time

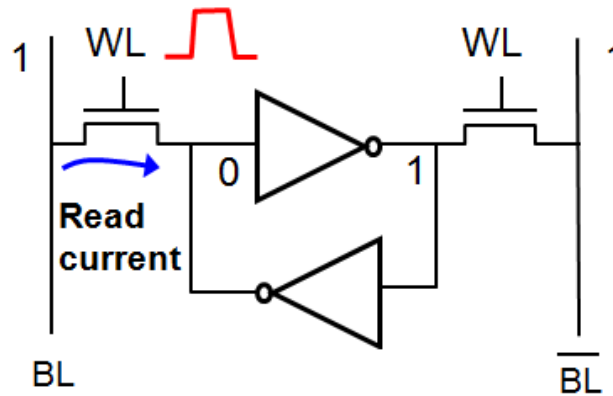


Fig.12. Circuit Setup for Read Access Time Measurement

The circuit setup for read access time measurement is shown in Figure 12, in which,

$$W = [4 \quad 2.5 \quad 1.875] * FS.$$

Unlike read DNM, in read access time measurement, we monitor the time needed for BL reducing from VDD to 90% VDD, which can be captured by the sens-amplifier.

Yield comparison among SPICE Monte-Carlo simulation, full model building simulation and partial-model building simulations is presented in the Table IV.

Table IV. Read Access Time Measurement in One Million Monte-Carlo Samples for Various Target Values

Critical time value	Failures from HSPICE	Failures from full Kriging model
1 ns	847	802
0.1 ns	1750	2236
0.05 ns	4635	5371

D. Boundary Convergence in Model Building Process

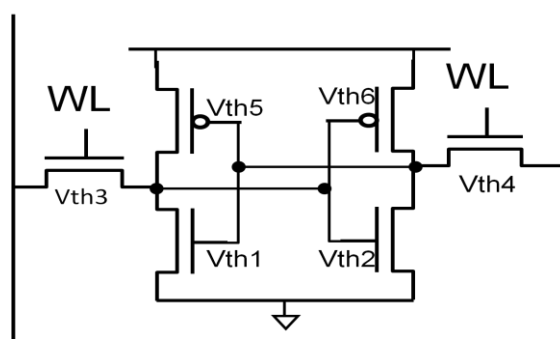


Fig.13. 6T SRAM Cell and Vth Assignments

Take write DNM as an example, the Vth assignments is as marked in Figure 13.

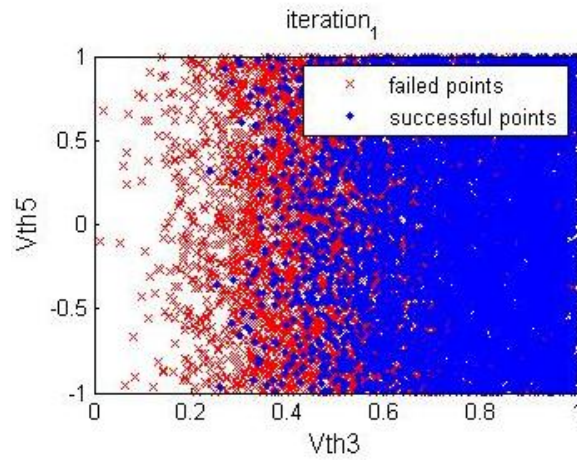
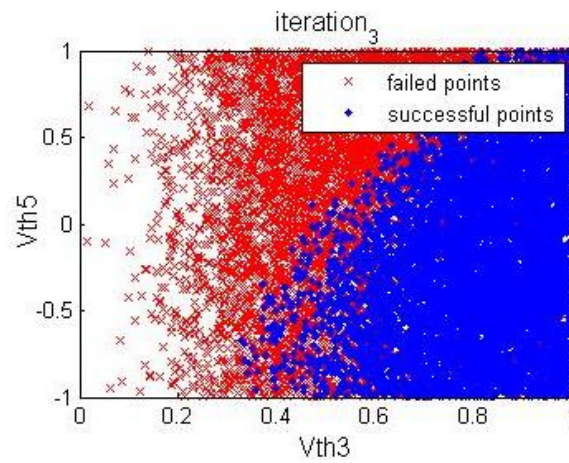
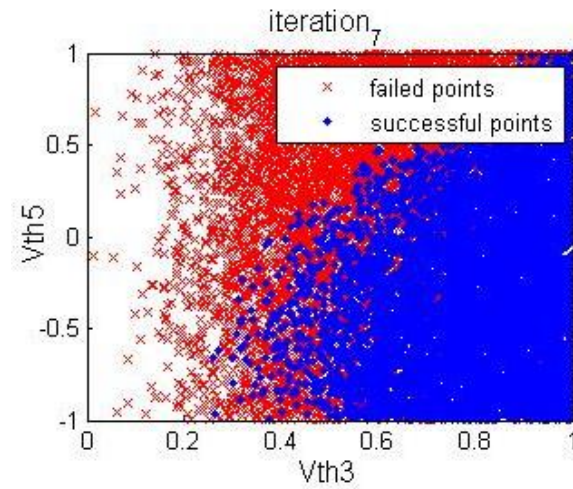
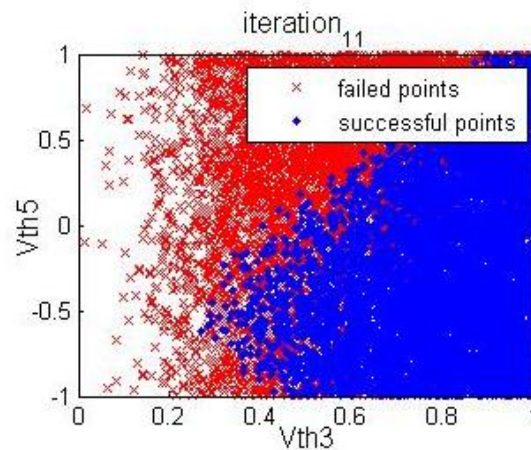
(a) The 1st Iteration(b) The 3rd Iteration

Fig.14. The Boundary Figuring Process

(c) The 7th Iteration

(d) The Converged Boundary

Fig.14 continued

From Figure14, we can see the converging process of the model which is as expected in Figure 7. In Figure 14(a) the boundary is roughly formed, but still with a lot of wrong discriminations. After series of iterations, the boundary is evolved to be clear

and accurately divided to two regions. The scale on the axis is not the real V_{th} value for each transistor. Instead, it is the variation percentage distance from the nominal V_{th0} .

The exact V_{th} is calculated as following:

$$V_{th_realsample} = V_{th0}(1 + V_{th}) \quad (5.4)$$

where V_{th0} is the nominal threshold voltage which is the ideal value without any variation, $V_{th_realsample}$ is the sampled value around V_{th0} considering process variations and V_{th} is the scaled value shown in Figure 14.

E. Yield Sensitivity Analysis and Model Verification

1. The Accuracy of Partial Kriging Model

The sensitivity analysis is developed on the foundation of partial Kriging model. In other words, after the yield analysis is finished for the fixed design, we utilize the last set of training data input from the previous stage and build the model for the tuned design to predict their yield and sensitivity. This strategy, called partial Kriging model, saves us the time of repeating the iteration of training data update process for every design parameter respectively.

To ensure the applicability, we experiment on the accuracy of this method by tuning back one of the design parameter by 5%. We build full Kriging model on the biased design and directly use the last set of selected training data input to build partial model on the fixed design. Therefore, we can compare the results with full Kriging model for the same fixed design as well as HSPICE simulation in the following Table V.

Table V. Accuracy of Partial Kriging Model

	Failure rate by HSPICE(per million)	Failure rate by full Kriging method(per million)	Failure rate by partial Kriging method(per million)
Read DNM	134	143	151
Write DNM	832	871	753
Read SNM	149	158	163
Write SNM	45	67	82
Read access time	847	802	765

2. Sensitivity Analysis

We use the proposed method to do sensitivity analysis and compare one set of the results with SPICE simulation. Table VI presents the experimental failure rate measurements. Based on that, the calculated yield sensitivity value with respect to each design parameter is shown in the “S” columns. The unit of the sensitivity is the yield change per one million samples per nanometer. The nominal design parameters used for measurement in each metric is the same as before in the yield analysis.

The unit of the sensitivity is the change in number of eligible cells per million per nanometer. We compared the Read DNM model results with HSPICE, although the corresponding sensitivity results seems to be a little different, considering the sample size is one million, in practice, the deviation is relative small and the accuracy level is satisfactory.

It has been demonstrated in [6] that a stronger access transistor (larger W_{ax}) increase the failure probability of read DNM due to the induced high voltage from bitline. However, it decreases the failure rate constrained by access time. Increasing W_n increases the strength of pull-down NMOS transistors, which leads to the reduction of failure rate in Read DNM and Access time limitation.

For SNM, increasing the size of the pull-down devices of the standard 6T SRAM cells for enhanced read stability comes at the cost of degraded write margin[21], which is also represented in the results.

The result is in accord with the description above, serving as a good verification of our model.

Table VI. Failure Rate Measurement and Yield Sensitivity Analysis for One Million Samples from Kriging Model and HSPICE

	Failures of the fixed (nominal) design	$\Delta W_{ax}=+5\%$		$\Delta W_n=+5\%$		$\Delta W_p=+5\%$	
		Failures	S_{ax}	Failures	S_n	Failures	S_p
Read DNM by model	151	423	-64.47	117	3.78	90	10.84
Read DNM by HSPICE	134	587	-107.38	93	4.56	52	14.58
Write DNM by model	753	539	13.59	733	2.22	1231	-84.98
Read SNM by model	163	234	-16.83	125	4.22	266	-18.31
Write SNM by model	82	59	2.04	131	-5.44	90	-1.42
Read access time by model	847	536	73.72	720	14.11	869	-3.91

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

In this thesis, we propose a statistical method for yield analysis and prove its accuracy and efficiency in SRAM design, both theoretically and practically. We first discuss the justification of utilizing Kriging Model to figure out the binary decision boundary of SRAM performance, and then develop a set of sequential updating strategies to update the model iteratively so as to make the model suitable for the circumstance of SRAM yield analysis. Furthermore, we conduct the sensitivity analysis based on the methodology of yield analysis, in endeavor to provide the information of how design parameters impact on different aspects of SRAM performance.

The model can be treated as a statistical simulator for yield analysis, regardless of sampling method in the evaluation stage. In the future work, it can be improved by co-operating with other efficient sampling method as described in [3, 4] to obtain more significant speedup in SRAM yield analysis.

REFERENCES

- [1] K. Chakraborty and P. Mazumder, *Fault-Tolerance and Reliability Techniques for High-Density Random-Access Memories*. Upper Saddle River, NJ: Prentice Hall, 2002.
- [2] B. Mohammad, M. Saint-Laurent, P. Bassett, and J. Abraham, "Cache design for low power and high yield," in *Proc. 9th International Symposium on Quality Electronic Design: IEEE Computer Society*, 2008, pp. 103-107.
- [3] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events," in *Proc. Design Automation Conference: ACM/IEEE*, 2006, pp. 69-72.
- [4] A. Singhee and R. A. Rutenbar, "Statistical blockade: a novel method for very fast Monte-Carlo simulation of rare circuit events, and its application," in *Proc. Design, Automation & Test in Europe Conference & Exhibition*, 2007, pp. 1-6.
- [5] A. Pavlov and M. Sachdev, *CMOS SRAM Circuit Design and Parametric Test in Nano-scaled Technologies: Process-aware SRAM Design and Test*. New York, NY: Springer, 2008.
- [6] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions*, vol. 24, pp. 1859-1880, Dec. 2005.

- [7] M. Khellah, Y. Ye, N. S. Kim, D. Somasekhar, G. Pandya, A. Farhang, K. Zhang, C. Webb, and V. De, "Wordline & bitline pulsing schemes for improving SRAM cell stability in low-V_{cc} 65nm CMOS designs," in *Proc. Symposium on VLSI Circuits: Digest of Technical Papers*, 2006, pp. 9-10.
- [8] S. Roy, B. Cheng, and A. Asenov, "Impact of intrinsic parameter fluctuations on circuits ": Device Modeling Group, University of Glasgow.
- [9] W. Dong, P. Li, and G. M. Huang, "SRAM dynamic stability: theory, variability and analysis," in *Proc. IEEE/ACM International Conference on Computer-Aided Design: ICCAD*, 2008, pp. 378-385.
- [10] H. Gunes, H. E. Cekli, and U. Rist, "Data enhancement, smoothing, reconstruction and optimization by Kriging interpolation," in *Proc. Simulation Conference, 2008. WSC 2008. Winter, 2008*, pp. 379-386.
- [11] H. You, M. Yang, D. Wang, and X. Jia, "Kriging model combined with Latin hypercube sampling for surrogate modeling of analog integrated circuit performance," in *Proc. 10th Int'l Symposium on Quality Electronic Design: ISQED*, 2009, pp. 554-558.
- [12] B. Ankenman, B. L. Nelson, and J. Staum, "Stochastic Kriging for simulation metamodeling," in *Proc. Simulation Conference, 2008. WSC 2008. Winter, 2008*, pp. 362-370.
- [13] T. J. Santner, B. J. Williams, and W. Notz, *The Design and Analysis of Computer Experiments*. New York, NY: Springer, 2003.

- [14] A. Gupta, Y. Ding, L. Xu, and T. Reinikainen, "Optimal parameter selection for electronic packaging using sequential computer simulations," *ASME Transactions, Journal of Manufacturing Science and Engineering*, vol. 128(3), pp. 705-715, 2006.
- [15] Z. Bin, A. Arapostathis, S. Nassif, and M. Orshansky, "Analytical modeling of SRAM dynamic stability," in *Proc. Computer-Aided Design, 2006. ICCAD '06. IEEE/ACM International Conference*, 2006, pp. 315-322.
- [16] J. Sacks, W. J. Welch, J. M. Toby, and H. P. Wynn, "Design and analysis of computer experiments," *Statistical Science*, vol. 4, pp. 409-423, 1989.
- [17] Y. Guo and L. Peng, "Yield-aware analog integrated circuit optimization using geostatistics motivated performance modeling," in *Proc. Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference*, 2007, pp. 464-469.
- [18] S. N. Lophaven, H. B. Nielsen, and J. Sondergaard, "Aspects of the Matlab toolbox DACE," *Informatics and Mathematical Modeling*, Lyngby, Copenhagen, Technical University of Denmark, DTU, Technical report 2002.
- [19] J. P. C. Kleijnen, "An overview of the design and analysis of simulation experiments for sensitivity analysis," *European Journal of Operational Research*, vol. 164, pp. 287-300, 2005.
- [20] J. C. Helton, J. D. Johnson, C. J. Sallaberry, and C. B. Storlie, "Survey of sampling-based methods for uncertainty and sensitivity analysis," *Reliability Engineering & System Safety*, vol. 91, pp. 1175-1209, 2006.

- [21] S. A. Tawfik and V. Kursun, "Stability enhancement techniques for nanoscale SRAM circuits: A comparison," in *Proc. SoC Design Conference, 2008. ISOCC '08. International*, 2008, pp. I-113-I-116.

VITA

Name: Chang Zhao

Address: 214 Zachary Engineering Center, College Station, TX 77843-3128

Email Address: zhao329@neo.tamu.com

Education: B.S., Electrical Engineering, Huazhong University of Science and Technology, 2007