Approximate Yield Estimation of Correlated Failure Events for Near-threshold SRAM

Zhongmao Sun*, Shuaibo Huang and Hao Yan

National ASIC System Engineering Research Center, Southeast University, No.2 Sipailou, Nanjing, Jiangsu Province, China

*E-mail: szm@seu.edu.cn

Abstract. In advanced manufacturing technology, SRAM yield estimation with correlation is a significant challenge due to process variations and structures especially at low voltage. In this paper, Double Asymptotic Probability Approximation (DAPA) method with Subset Analytical Model (SAM) is proposed to effectively estimate the probability of correlated failure event. SAM models SRAM failure events based on simulation of subsets of the equivalent circuit. DAPA obtains the recursive function of partial failure rate. The key idea is to approximate the failure rate of the entire system based on a linear combination of partial failure rates by decoupling failure correlation. Under TSMC 28 nm process, \((M=N=8)\) SRAM failure rate is estimated at 500 mV with an error of less than 5% by the proposed method. Besides 2.5 hours of modeling consumption, the calculation can be completed in 7 seconds, which greatly reduces SRAM failure estimation time, compared with Monte Carlo method.

1. Introduction

The on-chip SRAM is commonly used in SoC systems and consumes a significant fraction of the total chip power. For high integration density, SRAM has always been designed with the advanced integrated circuit (IC) technology which brings about serious process variations, making SRAM bitcells extremely vulnerable to failures especially at low voltage caused by random dopant fluctuations, litho-induced variations, oxide thickness variations, etc. However, as the supply voltage approaches the threshold voltage of transistors in CMOS technology, energy efficiency is near to the optimal point [1]. And near-threshold SRAM is widely applied in low-power embedded systems and IoT platforms. Thus, to achieve required or aggressive performance and power, it is contradictory to guarantee sufficient yield. Fortunately, some applications (e.g., neural networks) have been shown to be inherently error resilient, which may perform under the presence of bit errors to further improve energy-efficiency based on effective yield estimation.

Statistical methods addressing yield problems have been proposed in the literature. A widely used approach is Monte Carlo (MC) method based on randomly selected samples in the variation space. However, MC typically requires millions, or even billions of samples to reach reasonable accuracy, which is prohibitively costly. Therefore, other approaches based on importance sampling (IS) [2] and boundary-searching [3] are developed. The basic idea of IS is to sample the failure regions with distorted probability density function. And boundary-based methods are to construct the boundary of the failure regions. However, these approaches can only deal with low-dimensional variation spaces. Subset Simulation (SUS) [4] is proposed to handle the high-dimensional problem, which estimates the failure rate by multiplying the probabilities of several intermediate failure events.
The aforementioned methods only focus on the cell-level failure events (i.e., single bitcell failure) which cannot reflect the actual failure rate of entire SRAM. However, it is unrealistic to run Monte Carlo simulations of a large SRAM for the computational complexity and time consumption. Failure rate estimation of entire system has not been intensively studied yet. The difficulty is that system-level failure events are determined by a large number of cell-level failures. And the cell-level failures are often statistically correlated due to circuit structures [5], global process variations [6], supply voltage variations, etc. However, most existed methods assume that cell-level events are independent. In this case, ignoring or inappropriate simplifying correlation will lead to significant inaccuracy in yield estimation for relatively large SRAM arrays [7]. Analytical model [8] reflects the correlation among bitcell failures caused by the shared peripherals. But, performance parameters are modeled as Gaussian distributions which may not be consistent with actual distributions in advanced manufacturing technologies and at low voltage [9]. A novel Asymptotic Probability Approximation (APA) [10] is proposed to estimate system-level failure rate without distribution assumption that is approximated as a linear combination of partial failure rates. And the partial rates are calculated by Hierarchical Subset Simulation (H-SUS) method. But, the variation of sense amplifier enable time is not considered.

In this paper, the system-level failure events with correlation among bitcells are estimated with Double Asymptotic Probability Approximation (DAPA) and Subset Analytical Model (SAM). DAPA method is to decouple the system-level failure into a set of column-level, line-level and partial failure events. Then system-level failure rate is approximated as a linear combination of column-level, line-level and partial failure rates in two different ways. And it assumes they are approximately equal which can obtain the recursive function of partial failure rate. That is very different with APA. To make DAPA of practical utility, SAM is established by subsets of the equivalent circuit simulation and further decouples the correlation among bitcells. And Look-up Table (LUT) is utilized to record the relationship among performance parameters. Additionally, subsets of the equivalent circuit are not very dependent on system circuit, which means the simulation experiments can be done beforehand.

2. Background

2.1. SRAM Failure Model

Process parameter variations result in the mismatch in the strengths of different transistors in a single SRAM cell, which leads to functional failures (read, write, access and hold failure) [11]. Assume that the process variations are described by the vector $\mathbf{x}$ of a series of independent random variables and $y^{\text{cell}}(\mathbf{x})$ denotes the performance of interest (PoI) of a single SRAM bitcell. The cell-level failure rate can be expressed as:

$$p^{\text{cell}} = \Pr\left[y^{\text{cell}}(\mathbf{x}) \in F^{\text{cell}}\right]$$  \hspace{1cm} (1)

where $F^{\text{cell}}$ represents the cell-level failure region, and $\Pr[\cdot]$ represents the probability of an event [10]. Traditionally, $F^{\text{cell}}$ is estimated by Monte Carlo (MC) simulation by studying cell-level PoI with $N$ random samples, $\mathbf{x}^{(i)} (i=1,2,...,N)$, generated by its probability density function (PDF):

$$p^{\text{cell}} = \frac{1}{N} \sum_{n=1}^{N} I^{\text{cell}}\left[y^{\text{cell}}(\mathbf{x}^{n})\right]$$  \hspace{1cm} (2)

where $y^{\text{cell}}(\mathbf{x}^{n})$ stands for the value of cell-level PoI on the sample $\mathbf{x}^{n}$, and $I^{\text{cell}}[\cdot]$ is the indicator function. For instance, the read access failure can be modeled as the statistical relation among sense amplifier (SA) offset $V_{\text{OS}}$, sense amplifier enable time $T_{\text{SAE}}$, and the voltage difference between complementary bitlines $\Delta V_{\text{BL}}$ [8]. The dominant reason for read access failure is that a large $V_{\text{OS}}$ or a
short $T_{SAE}$ makes $\Delta V_{Bl}$ smaller than $V_{os}$. Moreover, the parameters of failure model are all unknown distributions due to process variations, especially at low supply voltage. Thus, the $I^{cell}(\cdot)$ of read access failure is:

$$I^{cell}(y^{cell}(x)) = \begin{cases} 1 & \Delta V_{Bl}(t) < V_{os} \\ 0 & \Delta V_{Bl}(t) \geq V_{os} \end{cases}$$

(3)

where $t$ is drawn from PDF of $T_{SAE}$. The failure mechanism becomes complex, as the $\Delta V_{Bl}(t)$ is an uncertain distribution at any time. And the dimension of parameter space will increase, as all the transistors are supposed to be taken into consideration. Thence, a meaningful failure estimation requires millions, or billions of samples to capture effective failure regions. In theory, to estimate the probability $P$ with $(1 - \epsilon)$ accuracy and $(1 - \delta)$ confidence, according to [12], the required sampling number of MC method is:

$$NUM_{MC} = \frac{\log(1/\delta)}{P \cdot \epsilon^2}$$

(4)

2.2. Correlated Failure Events

The system-level failure events are determined by a large number of cell-level failures. And the cell-level failures are often statistically correlated due to circuit structures [5], global process variations [6], supply voltage variations, etc. However, most existed methods assume that cell-level events are independent. In this case, ignoring or inappropriate simplifying correlation will lead to significant inaccuracy in yield estimation for relatively large SRAM arrays [7].

SRAM array is shown in Figure 1, which consists of $N$ columns with a shared replica bitline circuit. And each column consists of $M$ bitcells that share the same SA. Before a read operation, complementary bitlines ($BLs$ & $BLBs$) have been precharged to supply voltage and wait for read enable signal. Then corresponding word line $WL[0]$ and $RWL$ are enabled and the $RBLB$ begins to discharge through the replica cell. At the same time, bitlines are discharging through the selected bitcells and the $\Delta V_{Bl}$ is increasing. If SAE is enabled early when the developed $\Delta V_{Bl}$ is smaller than the SA offset voltage, a read access failure occurs and bitcells of the same line may fail. Additionally, if $V_{os}$ is relatively higher, bitcells of the same column are highly probable to fail. Notice that due to the shared SA and replica bitline circuit, cell-level failure events are correlated.

![Figure 1. SRAM array with conventional replica bitline technique.](image)

Furthermore, the distribution of $T_{SAE}$ is non-Gaussian [9] with a long tail because of weak bitcells at near-threshold voltages. If SAE is not enabled during word line enable time, the system-level failure occurs as well. Therefore, it is contradictory to guarantee sufficient yield to achieve required performance.
3. Proposed Approach

3.1. Problem Definition
In this work, we focus on estimating near-threshold SRAM yield based on error resilient applications as power or performance is susceptible to process variations at low voltage. And we are not faced with the rare failure event \(10^{-8} \sim 10^{-6}\) problem.

Suppose that we have a system containing \(N\) columns with a shared replica bitline circuit. And each column consists of \(M\) bitcells that share the same SA. The system-level failure event occurs if any one of the bitcells fails. Let \(x\) denotes its process-level random variables and involves global and local variations. The system-level PoI can be expressed as \(y(x) = [y_{11}(x)...y_{1N}(x)...y_{M1}(x)...y_{MN}(x)]\), where \(y_{MN}(x)\) means the PoI of the \(N\)-th bitcell of \(M\)-th line. If PoI of any bitcell does not satisfy the specification, system fails. To describe system-level failure events and the correlation issues, the conception of set is introduced. The system-level failure event can be expressed as a union of all cell-level failure events and the failure rate \(P^{ARR}\) can be expressed as:

\[
P^{ARR} = Pr[y(x) \in P^{ARR}] = Pr[cell_{11} \cup ... \cup cell_{1N} \cup ... \cup cell_{M1} \cup ... \cup cell_{MN}]
\]

Due to the correlation of cell-level failure events, \(cell_{ij} \cap cell_{uv} \neq \emptyset\), where \(cell_{ij}\) represents cell-level failure event \(y^{cell_{ij}}(x) \in F^{cell_{ij}}\), and \(ij, uv \in \{11, 12, ..., MN\}\). But, the failure rate of different bitcells is regarded to be same as all bitcells perform the same function [10]. If we completely ignore the correlations of cell-level failures (i.e., assume the failure events of different cells are independent), system-level failure estimation can be calculated by binomial distribution:

\[
P^{ARR} = 1 - (1 - P^{cell})^{MN}
\]

However, this method will lead to significant inaccuracy especially when \(MN\) is large. To accurately estimate system yield, effective analytical models considering correlation are required.

3.2. Double Asymptotic Probability Approximation
We propose the Double Asymptotic Probability Approximation (DAPA) method to decouple the system-level failure into a set of column-level, line-level and partial failure events. Then system-level failure rate is approximated as a linear combination of column-level, line-level and partial failure rates in two different ways. The partial failure events are defined as:

\[
P^{mn} = Pr[cell_{11} \cup ... \cup cell_{1n} \cup ... \cup cell_{M1} \cup ... \cup cell_{mn}]
\]

where \(m < M\), \(n < N\), and \(P^m = P^{i1}\). The partial failure rate can also be expressed by column-level or line-level failure rates:

\[
P^{mn} = Pr[col_{1} \cup col_{2} ... \cup col_{n}] = Pr[line_{1} \cup line_{2} ... \cup line_{m}]
\]

where \(col_{n}\) and \(line_{n}\) are union of all cell-level failure events. Since the function of different bitcells, columns or lines is the same, the probability of partial failure events does not concern the actual bitcells, columns or lines. Thus, it can be expressed as:

\[
P^{mn} = P^{m1} + P^{n1-1} - Pr[col_{1} \cap (\bigcup_{i=2}^{n} col_{i})]
\]

With the similar derivation process, \(P^{mn-1}\) can be expressed by \(P^{mn-2}\) and other terms. Then, \(P^{mn}\) can be expressed as:
\[ P^{\text{mun}} = nP^{\text{mun1}} - \sum_{k=2}^{n} \Pr[\text{col}_k \cap \bigcup_{i=2}^{k} \text{col}_i] \]  
(10)

In this way, \( P^{\text{mun1}} \) can be represented by \( P^{\text{i}} \) and other terms [10]:

\[ P^{\text{mun1}} = mP^{\text{i}} - \sum_{k=2}^{m} \Pr[\text{cell}_k \cap \bigcup_{i=2}^{k} \text{cell}_i] \]  
(11)

Ultimately, \( P^{\text{mun}} \) can be expressed as:

\[ P^{\text{mun}} = n(mP^{\text{i}} - \sum_{k=2}^{m} \Pr[\text{cell}_k \cap \bigcup_{i=2}^{k} \text{cell}_i]) - \sum_{k=2}^{n} \Pr[\text{col}_k \cap \bigcup_{i=2}^{k} \text{col}_i] \]  
(12)

Similarly, the partial failure rate can also be expressed by line-level failure rates:

\[ P^{\text{mun}} = m(nP^{\text{i}} - \sum_{k=2}^{n} \Pr[\text{cell}_k \cap \bigcup_{i=2}^{k} \text{cell}_i]) - \sum_{k=2}^{m} \Pr[\text{line}_k \cap \bigcup_{i=2}^{k} \text{line}_i] \]  
(13)

Notice that \( P^{\text{Mun}} \) can also be expressed in the form of (12), (13). The approximate relationship is established by ignoring the terms of \( m \) and \( d \) in (12), where \( M, n \) and \( N, m \) are approximately equal since they all represent the upper bound of the actual failure rate. The recursive function can be expressed as:

\[ P^{\text{mun}} = M(N - n)(P^{\text{i}} - P^{\text{(n-1)i}}) + (M - m)(P^{\text{mun1}} - P^{\text{(m-1)mun1}}) + P^{\text{mun}} = \mathcal{R}^{\text{mun}} \]  
(14)

In the same way, \( P^{\text{Mun}} \) can also be approximated by (13):

\[ P^{\text{Mun}} \leq M(N - n)(P^{\text{i}} - P^{\text{(n-1)i}}) + (M - m)(P^{\text{mun1}} - P^{\text{(m-1)mun1}}) + P^{\text{mun}} = \mathcal{R}^{\text{mun}} \]  
(15)

It can be mathematically proven that \( Q^{\text{mun}} \) and \( R^{\text{mun}} \) are upper bounds of the actual failure rate. The relationship between \( Q^{\text{mun}} \) and a subset of \( Q^{\text{mun}} \) or between \( R^{\text{mun}} \) and a subset of \( R^{\text{mun}} \) can be easily determined by applying set theory.

\[ P^{\text{mun}} \leq Q^{\text{mun}} \leq Q^{\text{mun}}(n-1), P^{\text{mun}} \leq R^{\text{mun}} \leq R^{\text{mun}}(n-1) \]  
(16)

Assume that \( Q^{\text{mun}} \) and \( R^{\text{mun}} \) are approximately equal since they all represent the upper bound of the actual failure rate. \( P^{\text{mun}} \) can be expressed as:

\[ P^{\text{mun}} = \frac{M}{1-t} (P^{\text{i}} - P^{\text{(n-1)i}}) - \frac{tN}{1-t} (P^{\text{mun1}} - P^{\text{(m-1)mun1}}) + \frac{P^{\text{mun1}} - P^{\text{(m-1)mun1}}}{1-t} \]  
(17)

where \( t = (M - m) / (N - n) \). \( P^{\text{(n-1)i}} \) and \( P^{\text{(m-1)mun1}} \) can be expended in this way. At length, \( P^{\text{mun}} \) is expressed by a set of column-level and line-level failure rates. To estimate \( P^{\text{(n-1)i}} \) and \( P^{\text{(m-1)mun1}} \), the
simultaneous failure rate $P_s^{(m-j)}$ and $P_s^{(n-j)}$ are introduced which mean the probabilities of all bitcells fail at the same time. $P_s^{(m-j)}$ and $P_s^{(n-j)}$ can be written as [10]:

$$
P_s^{(m-j)} = \sum_{k=1}^{m-j} (-1)^{k-1} \binom{m-j}{k} P_s^{(m-j)} , P_s^{(n-j)} = \sum_{k=1}^{n-j} (-1)^{k-1} \binom{n-j}{k} P_s^{(n-j)}
$$

(18)

Hence, the remaining problem is how to effectively estimate $P_s^{(m-j)}$ and $P_s^{(n-j)}$.

### 3.3 Subset Analytical Model

The Subset Analytical Model (SAM) is proposed to solve the remaining problem. According to the definition of $P_s^L$, where $L$ represents any size of column or line, $P_s^L$ can be written in the form of conditional probability:

$$
P_s^L = \Pr[\text{cell}_1 \cap \text{cell}_2 \ldots \cap \text{cell}_L] = \prod_{i=2}^L \Pr[\text{cell}_i | \text{cell}_1, \ldots, \text{cell}_{i-1}]
$$

(19)

$P_s^L$ is the joint probability of $L$ bitcells failure event and cell-level failure events are correlated due to the shared SA and replica bitline circuit. To address this correlation issue specifically, the read access failure event is taken as an example. From (3), we will further decouple the correlation among bitcells and $P_s^L$ can be rewritten as:

$$
P_s^L = \Pr[\Delta V^L_{BL}, V^L_{OS}, \ldots, V^L_{BL}, V^L_{OS}, T_{SAE}] = \Pr[\Delta V^L_{BL}, V^L_{OS}, \ldots, V^L_{BL}, V^L_{OS} | T_{SAE}] \Pr[T_{SAE}]
$$

(20)

The cell-level failure model is established with three parameters ($\Delta V_{BL}$, $V_{OS}$, $T_{SAE}$). According to the circuit structure shown in figure 1, each column shares the same SA and replica bitline circuit. However, each line only shares the same replica bitline circuit. Thus, column-level failure events depend on $V_{OS}$ and $T_{SAE}$. Line-level failure events only depend on $T_{SAE}$. $\Delta V_{BL}$ is independent among different bitcells and $V_{OS}$ is independent in line-level failure events. $P_s^{(m-j)}$ and $P_s^{(n-j)}$ can be rewritten in different ways as:

$$
P_s^{(m-j)} = \Pr[\Delta V^L_{BL}, \ldots, \Delta V^m_{BL}, V_{OS}, T_{SAE}] = \Pr[\Delta V^L_{BL}, \ldots, \Delta V^m_{BL} | V_{OS}, T_{SAE}] \Pr[V_{OS}] \Pr[T_{SAE}]
$$

(21)

$$
P_s^{(n-j)} = \Pr[\Delta V^L_{BL}, V^L_{OS}, \ldots, \Delta V^n_{BL}, V^n_{OS}, T_{SAE}] = \Pr[\Delta V^L_{BL}, V^L_{OS}, \ldots, \Delta V^n_{BL}, V^n_{OS} | T_{SAE}] \Pr[T_{SAE}]
$$

(22)

The conditional probability $\Pr[\text{cell}_i | \text{cell}_1, \ldots, \text{cell}_{i-1}]$ in (19) is on the condition of $T_{SAE}$ and $V_{OS}$ or just $T_{SAE}$ in different cases analytically. To calculate the conditional probability, plenty of samples of $\Delta V_{BL}$, $V_{OS}$ and $T_{SAE}$ are generated by simulation and Look-up Table (LUT) is utilized for storing and managing the data. This process is not complicated since we only simulate the subsets of the equivalent circuit. The samples can be applied to all of the bitcells as they perform on the same function and we assume global variations are the same value. Let us start from the calculation of $\Pr[\text{cell}_i | \text{cell}_1]$. When the samples ($\Delta V_{BL}$, $V_{OS}$, $T_{SAE}$) of $\text{cell}_1$ are input to the failure model, the samples of $T_{SAE}$ and $V_{OS}$ which make $\text{cell}_1$ fail, are selected. For column-level failure events, the selected samples ($T_{SAE}$, $V_{OS}$) are combined with $\Delta V_{BL}$ of $\text{cell}_i$, itself are input to the failure model. Then we will get the failure rate of $\text{cell}_2$ on the condition that $\text{cell}_1$ fails. Notice that $\Delta V_{BL}$ is supposed to be searched in the LUT at different $T_{SAE}$. Then $\Pr[\text{cell}_i | \text{cell}_1, \ldots, \text{cell}_{i-1}]$ can be calculated in this way. For improving the calculation effectively, matrix is used. For instance, the discrete joint probability distribution of line-level of failure events at a determinate $T_{SAE}$ can be expressed as:
where \( bl_{m} \) is one of the samples \( \Delta V_{BL} \), and \( os_{n} \) is one of the samples \( V_{OS} \). As we can see, if we want to guarantee the sufficient accuracy, large numbers of samples are required which may consume a lot of storage resources. To solve this problem, samples are quantized for different accuracy requirements and resampled at failure regions.

Process parameter space is mapped into performance parameter space by SAM without any distribution assumption. Moreover, SAM is established by subsets of the equivalent circuit simulation, which consumes less time. Additionally, subsets of the equivalent circuit are not very dependent on system circuit. Namely, the simulation experiments can be done beforehand.

4. Implementation Details

4.1. Subset Analytical Model

The crucial conception of Subset Analytical Model is to find the dependent variables in the performance parameter space and decouple the correlation among bitcells. It makes DAPA of practical utility by giving the results of \( P_{s}^{(m-j)} \) and \( P_{s}^{(n-j)} \) in this procedure. Algorithm 1 summarizes this method.

**Algorithm 1:** Subset Analytical Model (SAM)

1. First, subsets of the equivalent circuit for required design are built.

2. Run simulations and obtain the discrete distributions of \( \Delta V_{BL} \{bl_{j}; i=1,2,...,n, j=t_{1},t_{2},...,t_{m}\} \), \( V_{OS} \{os_{i};i=1,2,...,n\} \) and \( T_{SAE} \{t_{i};i=1,2,...,n\} \).

3. Quantize the discrete sequences of \( \Delta V_{BL} \) and \( V_{OS} \) (Quantization error is from \( 10^{-3} \) to \( 10^{-6} \)).

4. Quantize the discrete sequences of \( T_{SAE} \{t_{1},t_{2},...,t_{n}\} \).

5. For \( t=t_{1},t_{2},...,t_{m} \)

   - Search the LUT and obtain the samples of \( \Delta V_{BL} \) at the time of \( t \).
   - Keep the different samples and calculate the frequency of the same samples.
   - Find the overlapping region of \( \Delta V_{BL} \) and \( V_{OS} \).
   - Resample only at the overlapping region.
   - Calculate \( \Pr[cell_{i},SAE]=\Pr[cell_{i}]+\Pr[T_{SAE}(t)]\Pr[(bl_{1}(t),os_{1}),...(bl_{m}(t),os_{n})] \) based on (23).
   - Select the samples of \( V_{OS} \) and \( T_{SAE} \) that make cell\(_{i}\) fail.

   End for

6. Calculate \( P_{s}^{(m-j)} \) and \( P_{s}^{(n-j)} \) according to (19), (21), (22).

Subsets of the equivalent circuit contain bitcell, SA and replica bitline circuit. Each of them can be simulated individually with corresponding capacitances according to the extraction of post-simulation or by experience. It is necessary to generate sufficient samples to abstract the failure model from process variations and this method do not need to base on distribution assumption. The relationship
between $\Delta V_{\text{BL}}$ and $T_{\text{SAE}}$ is based on LUT.

Because of limited memory, quantization and resampling are quite effective in step 3 and step 5. When the quantization error is less than $10^{-6}$, there is almost no loss in accuracy. However, accuracy may not be accepted if the granularity of quantization is too rough.

In detail, to compare the value between $\Delta V_{\text{BL}}$ and $V_{\text{OS}}$, $\Delta V_{\text{BL}}$ multiplies the reciprocal of $V_{\text{OS}}$ in the form of matrix (23) and it is easy to identify the values less than 1. These samples are in the failure regions. Then the conditional probabilities can be calculated and this procedure do not need to generate extra samples by simulation.

### 4.2. Double Asymptotic Probability Approximation

Double Asymptotic Probability Approximation method can be utilized to estimate system-level failure events based on SAM. This method calculates the upper bounds of $P_{MN}$ twice and assumes they are approximately equal. $P_{MN}$ can be approximately expressed by a set of column-level or line-level failure events eventually.

As $P_{mn}$ is a recursive function (17), the result depends on its subset. However, $P_{s}^{(m-1)}$ and $P_{s}^{(n-1)}$ are already given by SAM. Thus, calculation can be step by step. We will compare the accuracy of $P_{MN}$ under different $(m, n)$ according to (14), (15).

#### Algorithm 2: Double Asymptotic Probability Approximation (DAPA)

1. Load $P_{s}^{(m-1)}$ and $P_{s}^{(n-1)}$ calculated by SAM.
2. Calculate $P_{s}^{(m-1)}$ and $P_{s}^{(n-1)}$ according to (18).
3. For $m = 2, 3, \ldots (M-1)$
   - For $n = 2, 3, \ldots (N-1)$
     - If $(N-n)-(M-m) \neq 0$
       - Calculate $P_{mn}$ based on (17).
     - End if
   - End for
- End for
4. Calculate the upper bound of $P_{MN}$ under different $(m, n)$ according to (14), (15).

### 5. Numerical Experiments

The circuit in our experiments is shown in figure 1 and the bitline capacitances are all estimated 22 fF. Standard 6T SRAM cell, high-impedance input latch-type sense amplifier [13] and conventional replica bitline circuit [14] are adopted with TSMC 28nm HPC technology. The simulation experiments are designed for the read operation at 500 mV as we care more about the characteristics of near-threshold SRAM due to high energy efficiency.

It is unrealistic to run Monte Carlo simulations of a large SRAM for the computational complexity and time consumption. To verify our methods, M-by-N SRAM netlist is created $(M=N=8)$. We perform a large number of numerical Monte Carlo simulations in HSPICE to estimate SRAM yield.

#### 5.1. Subset Analytical Model

Single bitcell, SA and replica bitline circuit are simulated individually with corresponding capacitances $(22+\Delta)$ fF, where $\Delta$ is a correction term based on the extraction of post-simulation.
The number of simulation is $10^5$ and the cell-level failure rate $Pr(cell)$ is $6.697 \times 10^{-4}$. The golden cell-level failure rate is $6.678 \times 10^{-4}$ according to MC simulation. $P^{(m-j)_{1}}$ and $P^{(n-j)_{1}}$ are estimated by MC and PIE shown in figure 2. (a). PIE means that failure events are independent calculated by (6). $P^{(m-j)_{1}}$ and $P^{(n-j)_{1}}$ are shown in figure 2. (b) estimated by SAM and MC. The number of bitcells varies from 1 to 8. As can be seen, SAM matches the simulation results well.

Figure 2. (a) Failure rate estimations of column and line obtained by applying MC and PIE. (b) The simultaneous failure rate of column and line estimated by SAM and MC

5.2. Double Asymptotic Probability Approximation

$P^{MN}$ is approximated according to (14) or (15). We compared the accuracy under different $(m, n)$ shown in figure 3. The results of (2, 5) and (5, 2) approach the desired value well. But, when $(m, n)$ is not appropriate, the calculation will be not very accurate. Thus, the requirements of $(m, n)$ are analysed. If $m$ is close to $M$, $n$ is close to $N$ or $(M-m)$ is close to $(N-n)$, corresponding terms will be eliminated according to (14), (15) and (17), which may result in inaccurate results.

Table 1 gives the numerical comparisons among MC, PIE assumption and our method. Note that the number of simulation is not quite sufficient for $(M=N=8)$ SRAM. However, the time consumption cannot be acceptable already. Our proposed methods can be utilized to accurately estimate $(M=N=8)$ SRAM yield with 2.5 hours of modeling consumption. And we consider the model can be built beforehand. The calculation of SAM and DAPA almost consumes no time. The error is 3.4% compared with Monte Carlo method.

Figure 3. The failure rate of $P^{MN}$ approximated by DAPA under different $(m, n)$. PMN is estimated by MC. The results of (2, 5) and (5, 2) approach the desired value well.

| SRAM size $(M, N)$ | Number of simulation | Runtime of simulation | #MC | #PIE | #SAM | #DAPA | Consumption of method |
|-------------------|----------------------|-----------------------|-----|------|------|-------|-----------------------|
| Single bitcell    | $10^5$               | 2.5 hours             | $6.678 \times 10^{-4}$ | $6.678 \times 10^{-3}$ | $6.697 \times 10^{-3}$ | -- | 2.5 hours 2~5 second |
| (4, 4) array      | $10^5$               | 34 hours              | 0.0054 | 0.0106 | -- | 0.0049 | -- | 2 second |
| (6, 6) array      | $10^5$               | 76 hours              | 0.0099 | 0.0238 | -- | 0.0095 | -- | 2 second |
| (8, 8) array      | $10^5$               | 134 hours             | 0.0148 | 0.0419 | -- | 0.0153 | -- | 2 second |
6. Conclusion
In this paper, we propose Double Asymptotic Probability Approximation (DAPA) method with Subset Analytical Model (SAM) to effectively estimate the probability of correlated system failure events. SAM models SRAM failure events based on simulation of subsets of the equivalent circuit. This procedure, as we consider, can be done beforehand. Assume that the upper bounds of actual failure rate are approximately equal. DAPA obtains the recursive function of partial failure rate. The entire system failure rate is approximated based on a linear combination of partial failure rates by decoupling failure correlation. The requirements of partial failure events are analysed. Under TSMC 28 nm process, \((M=N=8)\) SRAM failure rate is estimated at 500 mV with an error of less than 5% by the proposed method. Besides 2.5 hours of modeling consumption, the calculation can be completed in 7 seconds, which greatly reduces SRAM failure estimation time compared with Monte Carlo method. In the future research, we will improve the model based on importance sampling.

Acknowledgments
This work was supported in part by the National Natural Science Foundation of China under Grant NSFC 61874152.

References
[1] Alioto M 2012 Ultra-low power VLSI circuit design demystified and explained: A tutorial IEEE Trans. Circuits Syst. I Reg. Papers vol 59 no.1 pp 3–29
[2] Shi X, Liu F Y, Yang J and He L 2018 A Fast and Robust Failure Analysis of Memory Circuits Using Adaptive Importance Sampling Method 55th ACM/ESDA/IEEE Design Automation Conference (DAC) IEEE pp 1-6
[3] Gong F, Yu H, Shi Y, Kim D, Ren J, and He L 2010 QuickYield: An efficient global-search based parametric yield estimation with performance constraints Design Automation Conference (DAC) pp 392–7
[4] Sun S and Li X 2014 Fast statistical analysis of rare circuit failure events via subset simulation in high-dimensional variation space Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design. IEEE pp 324-31
[5] Francis S, Kanj R, Joshi R, Kayssi A and Chehab A 2014 Statistical methodology for modeling non-IID memory fails events Fifteenth International Symposium on Quality Electronic Design. IEEE pp 205-211
[6] Wang J, Yaldiz S, Li X and Pileggi L T 2009 SRAM parametric failure analysis Proceedings of the 46th Annual Design Automation Conference ACM pp 496-501
[7] Zhang J and Gupta S 2014 SRAM array yield estimation under spatially-correlated process variation IEEE 23rd Asian Test Symposium IEEE pp 149-55
[8] Kang H, Kim J, Jeong H, Yang Y H and Jung S O 2015 Architecture-Aware Analytical Yield Model for Read Access in Static Random Access Memory IEEE Transactions on Very Large Scale Integration (VLSI) Systems vol 23 no.4 pp 752-65
[9] Balef H A, Kamal M, Afzali-Kusha A and Pedram M 2016 All-Region Statistical Model for Delay Variation Based on Log-Skew-Normal Distribution IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems vol 35 no.9 pp 1503-08
[10] Yu H, Tao J, Liao C, Su Y, Zhou D, Zeng X and Li X 2016 Efficient statistical analysis for correlated rare failure events via asymptotic probability approximation Proceedings of the 35th International Conference on Computer-Aided Design ACM p 18
[11] Mukhopadhyay S, Mahmoodi H and Roy K 2005 Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS IEEE transactions on computer-aided design of integrated circuits and systems vol 24 no.12 pp 1859-80
[12] Dolecek L, Qazi M, Shah D and Chandrakasan A 2008 Breaking the simulation barrier: SRAM evaluation through norm minimization Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design pp 322-9
[13] Zhu J, Bai N and Wu J 2013 A Review of Sense Amplifiers for Static Random Access Memory IETE Technical Review vol 30 no.1 pp 72-81
[14] Lin Z, Wu X, Li Z, Guan L, Peng C, Liu C and Chen J 2017 A pipeline replica bitline technique for suppressing timing variation of SRAM sense amplifiers in a 28-nm CMOS process IEEE Journal of Solid-State Circuits vol 52 no.3 pp 669-77