Rethinking Memory Redundancy: Optimal Bit Cell Repair for Maximum-Information Storage

Xin Li
Electrical & Computer Engineering Department, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
xinli@ece.cmu.edu

ABSTRACT
SRAM design has been a major challenge for nanoscale manufacturing technology. We propose a new bit cell repair scheme for designing maximum-information memory system (MIMS). Unlike the traditional memory repair that attempts to replace all failed bit cells by redundant columns and/or rows, we propose to repair the important bits (e.g., the most significant bit) only so that the information density (i.e., the number of information bits per unit area) is maximized. Towards this goal, an efficient statistical algorithm is derived to efficiently estimate the information density and then optimize the memory system for maximum-information storage. Our experimental results demonstrate that with a traditional 6-T SRAM cell designed in a commercial 45nm CMOS process, the proposed MIMS design can successfully operate at an extremely low power supply voltage (i.e., 0.6 V) and improve the signal-to-noise ratio (SNR) by more than 20 dB compared to the traditional SRAM design.

Categories and Subject Descriptors
B.7.2 [**Integrated Circuits**]: Design Aids – Verification

General Terms
Algorithms

Keywords
Integrated Circuit, Process Variation, Memory

1. INTRODUCTION
Static random access memory (SRAM) is a critical component for today's large-scale integrated systems [1]. An SRAM bit cell is often carefully designed to achieve: (1) nearly zero failure probability, and (2) maximal cell density. Since SRAM is extremely sensitive to large-scale process variations (e.g., random dopant fluctuations), it becomes increasingly difficult to design robust SRAM cells with today's nanoscale IC technology [1]-[10]. Hence, there is an immediate need to re-think the fundamental design strategy in order to meet today's manufacturing reality.

Recently, a new SRAM design methodology, referred to as maximum-information storage system, has been developed in [11]. The key idea is not to maximize the traditional cell density that is measured by the number of SRAM cells per unit area. Instead, the information density (i.e., the number of information bits per unit area) is maximized. The efficacy of maximum-information storage has been successfully demonstrated for several important applications. However, the design methodology proposed in [11] must use different SRAM cells with different transistor sizes. It, in turn, results in prohibitively high design cost, because designing and qualifying an SRAM cell requires a large amount of efforts with multiple silicon re-spins.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DAC'11, June 5-10, 2011, San Diego, California, USA
Copyright © 2011 ACM 978-1-4503-0636-2/11/06...$10.00

Motivated by this observation, we propose a completely new design methodology for Maximum-Information Memory System (MIMS) by exploring bit cell redundancy. Unlike the traditional SRAM repair scheme that aims to replace all failed bit cells by redundant columns and/or rows, we propose to distinguish different bit cells according to their levels of importance. For instance, considering the application-specific cases such as signal processing and numerical computation, the most significant bit (MSB) is more important than the least significant bit (LSB) when storing numerical data. Hence, the MSB should be better protected to achieve a smaller failure rate than the LSB in order to minimize the information loss. In other words, we do *not* attempt to fix all failed bit cells, since a zero failure rate is not necessary in many practical applications. As will be demonstrated by an image processing example in Section 4, a number of "unimportant" bit cells (e.g., the LSB of a pixel) can fail to work and they have negligible impact on the overall signal-to-noise ratio (SNR). By focusing on the important bits, we can reserve less redundant cells and, hence, use more bit cells for information storage. It, in turn, increases the information bits that we can store within a unit area.

To make the proposed MIMS system of practical utility, a number of architecture-level parameters (e.g., the number of redundant columns and/or rows) must be optimally determined to maximize the information density. For instance, if too much redundancy is allocated for bit cell repair, only few bit cells (hence, only few information bits) can be stored within a given area. On the other hand, if too little redundancy is reserved, the failure of the important bits cannot be appropriately repaired and, therefore, maximum-information storage cannot be achieved either. The challenging issue here is how to *optimally* design memory redundancy with an optimal bit cell repair scheme for our proposed MIMS system.

In this paper, an efficient statistical algorithm is developed to quantitatively measure the information density for a given SRAM design where the bit cells are optimally repaired. Our proposed algorithm is derived from the theories of order statistics [14]. It allows us to optimally integrate memory redundancy for bit cell repair and then efficiently predict the information density of the entire memory system. This, in turn, enables us to quickly compare and optimize different memory architectures to maximize the information density for data storage.

In order to fully demonstrate the efficacy of the proposed MIMS system, an image processing example is extensively studied in Section 4. Our experimental results demonstrate that with a traditional 6-T SRAM cell designed in a commercial 45nm CMOS process, the proposed MIMS design can successfully operate at an extremely low power supply voltage (i.e., 0.6 V) and deliver a superior SNR (i.e., 26.9 dB). In this example, MIMS is

able to improve the SNR by more than 20 dB compared to the traditional SRAM design. It is expected that, as SRAM cells become more sensitive to process and environmental variations due to technology scaling, the benefit of the proposed MIMS system will be more pronounced in the near future.

2. BACKGROUND

Similar to [11], we consider an SRAM system that is particularly designed for the data cache of signal processing or numerical computation applications. Without loss of generality, we assume that the SRAM cells are used to store the real-valued data within the interval [0, 1]. Any real-valued number can be mapped to this interval after appropriate shifting and scaling. In this case, a numerical number x can be represented by a set of binary digits $\{x_n; n = 1,2,\ldots\}$ where $x_n \in \{0, 1\}$:

$$x = \sum_{n=1}^{+\infty} 2^{-n} \cdot x_n \quad (1)$$

In practice, we always use a finite number of (say, N) digits $\{x^Q_n; n = 1,2,\ldots,N\}$, where $x^Q_n \in \{0, 1\}$, to approximate x. The set of digits $\{x^Q_n; n = 1,2,\ldots,N\}$ represents a *word* and N denotes the *word length*. Such an approximation is referred to as quantization in digital signal processing [12]:

$$x^Q = \sum_{n=1}^{N} 2^{-n} \cdot x_n^Q \quad (2)$$

$$x = x^Q + \varepsilon \quad (3)$$

where x^Q represents the quantized value and ε denotes the quantization noise. The power of ε is equal to [12]:

$$E(\varepsilon^2) = \frac{1}{12} \cdot 4^{-N} \quad (4)$$

where $E(\bullet)$ stands for the expected value of a random variable.

In addition to the quantization noise ε in (3), bit cell failure is another important noise source in today's SRAM system. When the binary digits $\{x^Q_n; n = 1,2,\ldots,N\}$ are stored in an SRAM, the stored value $\{y^Q_n; n = 1,2,\ldots,N\}$ can be different from $\{x^Q_n; n = 1,2,\ldots,N\}$, since each bit cell can possibly fail. In other words, the stored value:

$$y^Q = \sum_{n=1}^{N} 2^{-n} \cdot y_n^Q \quad (5)$$

can be different from the actual quantized value x^Q. The bit cell failure can be conceptually considered as an additive noise for the quantized value x^Q [11]:

$$x^Q = y^Q + \delta \quad (6)$$

where

$$\delta = \sum_{n=1}^{N} 2^{-n} \cdot \delta_n \quad (7)$$

stands for the "equivalent" noise caused by bit cell failure, and $\delta_n \in \{-1,0,1\}$ represents the error of the nth bit.

Combining (3) and (6) yields:

$$x = y^Q + \varepsilon + \delta . \quad (8)$$

Eq. (8) implies an important fact that when storing a numerical number x by N bits in an SRAM, two additive noise terms are introduced: (1) ε due to quantization error, and (2) δ due to bit cell failure. The total noise power is equal to [11]:

$$P_{NOISE} = E(\varepsilon^2) + E(\delta^2). \quad (9)$$

On the other hand, if the numerical number x is uniformly distributed over [0, 1], its energy is equal to [11]:

$$P_{SIG} = \frac{1}{12} . \quad (10)$$

Hence, the signal-to-noise ratio (SNR) is [11]:

$$SNR = -10 \cdot \log_{10}(12) - 10 \cdot \log_{10}\left[E(\varepsilon^2) + E(\delta^2)\right]. \quad (11)$$

It has been shown in [11] that *maximum-information storage can be achieved by maximizing the signal-to-noise ratio*. The aforementioned noise model is summarized in Figure 1.

To maximize the stored information, different strategies should be applied to minimize different noise sources. For example, to reduce the quantization noise ε, we should increase the word length (i.e., N) and, hence, the number of bit cells for information storage. On the other hand, to minimize the noise δ caused by bit cell failure, extra redundancy should be reserved to repair failed bit cells, thereby reducing the number of bit cells that can be used to store information. For this reason, the two noise terms ε and δ represent two conflicting performance metrics. For a given area, reducing ε will simultaneously increase δ, and vice versa. The open question here is how to optimally explore the trade-off between ε and δ to maximize the information that can be stored. In what follows, we will derive an information model that allows us to quickly calculate the SNR in (11) and efficiently explore the design trade-off for different memory architectures.

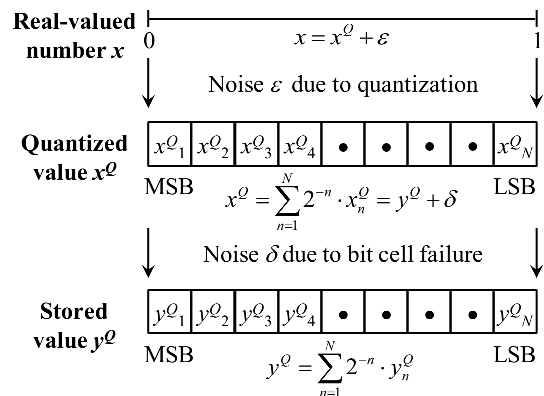


Figure 1. Two additive noise terms, ε due to quantization error and δ due to bit cell failure, are modeled when storing a real-valued number x.

3. MIMS DESIGN

The proposed MIMS design is facilitated by a combination of two new techniques: (1) an optimal bit cell repair scheme, and (2) an efficient algorithm for information modeling and optimization. In this section, we will describe these techniques in detail.

3.1 Optimal Bit Cell Repair

Figure 2 shows the block diagram of a typical SRAM array with M_R rows and M_C columns. For the traditional SRAM repair, if one column (or row) of the SRAM array contains failed bit cells, it will be replaced by another redundant column (or row) without bit cell failure. The objective of the traditional SRAM repair is to fix all failed bit cells by using redundancy. The repair is successful if and only if there are a sufficient number of redundant columns/rows to replace all failed bit cells. The traditional SRAM repair scheme does not distinguish the levels of importance for different bit cells. For example, when storing a numerical number, the MSB is much more important than the LSB. Hence, the MSB

should be better protected to achieve a smaller failure rate than the LSB in order to minimize the information loss.

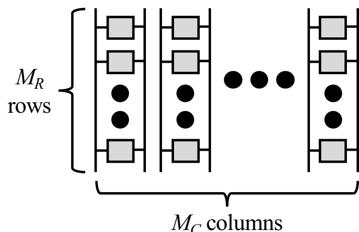


Figure 2. A typical SRAM array with M_R rows and M_C columns.

Motivated by this observation, we propose a completely different SRAM repair scheme in this paper. Given the SRAM array in Figure 2, we use each row to store a single quantized numerical number x^Q (i.e., a word). Different columns in the same row are used to store different bits of x^Q. To enable optimal SRAM repair, we count the number of failed bit cells $\{N_{Fj}; j = 1,2,\ldots,M_C\}$ for each column. Next, we sort all columns based on $\{N_{Fj}; j = 1,2,\ldots,M_C\}$. *The important bits of the numerical data should be stored in the columns with a small number of failed cells.* For instance, the MSB should be assigned to the column that is associated with the minimal value of $\{N_{Fj}; j = 1,2,\ldots,M_C\}$. As such, the information loss due to bit cell failure is minimized.

It is important to emphasize that the proposed SRAM repair scheme carries the following two important features.

- *Not all failed bit cells are repaired.* It is possible that there remain a number of failed bit cells, after the repair is made. Most likely, these failed cells do not represent the important bits of the numerical data. Hence, they will have negligible impact on the overall SNR.
- *Not all columns are used for data storage.* In other words, the word length N can be less than the number of columns M_C. If a column has too many failed bit cells, it cannot be used to store any useful information.

The percentage of the non-used columns can be conceptually viewed as a quantitative measure of memory redundancy. In most traditional SRAM designs, the redundancy is around 10%. For our proposed MIMS system, however, the optimal redundancy value depends on a number of other design parameters (e.g., the bit cell failure rate, the number of columns and rows, etc.), as will be demonstrated by our experimental results in Section 4.

To make the aforementioned SRAM repair of practical utility, statistical analysis is required to quantitatively assess the quality (e.g., the SNR) of a given memory system and then optimally determine the design parameters (e.g., the redundancy value) by comparing and optimizing different memory architectures to achieve maximum-information storage. In the next sub-section, we will develop a new statistical algorithm to address these issues.

3.2 Information Modeling and Design Optimization

As shown in (11), the SNR depends on two different terms: (1) $E(\varepsilon^2)$ due to quantization error, and (2) $E(\delta^2)$ due to bit cell failure. The power of the quantization noise can be simply calculated by (4). Hence, the major focus of this sub-section is to estimate the noise power $E(\delta^2)$.

Give (7), we have:

$$E\left(\delta^2\right) = E\left[\left(\sum_{n=1}^{N} 2^{-n} \cdot \delta_n\right)^2\right] \le E\left(\sum_{i=1}^{N}\sum_{j=1}^{N} 2^{-i-j} \cdot |\delta_i| \cdot |\delta_j|\right) \quad (12)$$

where $|\delta_i| \in \{0,1\}$ indicates whether the ith bit is failed ($|\delta_i| = 1$) or not ($|\delta_i| = 0$). Eq. (12) provides an upper bound of the noise power $E(\delta^2)$. In order to estimate $E(\delta^2)$ in (12), we must calculate the second-order statistics $E(|\delta_i|\cdot|\delta_j|)$ where $i,j \in \{1,2,\ldots,N\}$. Such statistical estimation is not trivial due to two reasons. First, all columns in the SRAM array are sorted according to the number of failed bit cells during memory repair. Such a sorting process can substantially change the failure rate of each bit cell. For example, since the MSB is assigned to the column that has the minimal number of failed bit cells, the "effective" failure rate of MSB should be much smaller than the failure rate of a bit cell without repairing. Second, it is well-known that the failure probability of an SRAM cell is extremely small (e.g., less than 10^{-5}). Such a small failure rate cannot be efficiently estimated by a brute-force Monte Carlo analysis, since an extremely large number of (e.g., 10^6~10^7) random samples are needed in order to accurately estimate the failure rate, thereby resulting in prohibitively high computational cost [4]-[9].

These observations motivate us to develop a new statistical analysis algorithm to efficiently calculate the second-order statistics $E(|\delta_i|\cdot|\delta_j|)$ in (12). Our proposed method is derived from the theories of order statistics [14]. It calculates $E(|\delta_i|\cdot|\delta_j|)$ by a number of analytical equations and, therefore, achieves orders of magnitude more efficiency than a brute-force Monte Carlo analysis.

A. Information Modeling

Since the proposed SRAM repair scheme maps each column of the memory array to one bit of the numerical data based on $\{N_{Fj}; j = 1,2,\ldots,M_C\}$ (i.e., the number of failed bit cells in each column), we need to first study the statistics of $\{N_{Fj}; j = 1,2,\ldots,M_C\}$. To this end, we define the following Bernoulli random variables $\{\xi_{ij}; i = 1,2,\ldots,M_R, j = 1,2,\ldots,M_C\}$ to model the failure events of the bit cells:

$$p\left(\xi_{ij}\right) = \begin{cases} \alpha & \left(\xi_{ij} = 1\right) \\ 1-\alpha & \left(\xi_{ij} = 0\right) \end{cases} \quad (13)$$

where $\xi_{ij} = 1$ indicates a failed bit cell at the ith row and jth column of the SRAM array, $p(\bullet)$ represents the probability of a random variable, and α is the failure probability of the bit cell (without repairing). In this paper, we assume that all bit cells share the same failure probability and their failure events are mutually independent, before the repair is applied. This assumption is valid, if the bit cell failure is mainly caused by random device mismatches, as is demonstrated by many previous works [2]-[10].

Given (13), the number of failed bit cells in the jth column is equal to:

$$N_{Fj} = \sum_{i=1}^{M_R} \xi_{ij} \ . \quad (14)$$

It is easy to verify that N_{Fj} in (14) satisfies binominal distribution. Its probability mass function (PMF) and cumulative distribution function (CDF) are equal to [13]:

$$pmf_{NF}(k) = \binom{M_R}{k} \cdot \alpha^k \cdot (1-\alpha)^{M_R-k} \quad \left(0 \le k \le M_R\right) \quad (15)$$

$$cdf_{NF}(k) = \sum_{i=0}^{k} \binom{M_R}{i} \cdot \alpha^i \cdot (1-\alpha)^{M_R-i} \quad \left(0 \le k \le M_R\right). \quad (16)$$

Since the probability distributions for $\{N_{Fj}; j = 1,2,\ldots,M_C\}$ are identical, we do not include the subscript "j" in (15)-(16). While

the distribution of N_{Fj} is easy to calculate, we need to further sort $\{N_{Fj}; j = 1,2,\ldots,M_C\}$, yielding:

$$N_{F(1)} \le N_{F(2)} \le \cdots \le N_{F(M_C)}. \quad (17)$$

In other words, given the random variables $\{N_{Fj}; j = 1,2,\ldots,M_C\}$, we sort their sample values in ascending order. The sorted random variables $\{N_{F(j)}; j = 1,2,\ldots,M_C\}$ are referred to as order statistics in the literature [14].

For our proposed MIMS system, $N_{F(j)}$ represents the number of failed bit cells for the jth column *after* the repair is made. The statistical distribution of $N_{F(j)}$ is completely different from that of N_{Fj} due to the sorting process applied during memory repair. Based on the theories of order statistics, the CDF of $N_{F(j)}$ can be written as [14]:

$$cdf_{NF(j)}(k) = \sum_{i=j}^{M_C} \binom{M_C}{i} \cdot cdf_{NF}^{i}(k) \cdot [1 - cdf_{NF}(k)]^{M_C - i} \quad (0 \le k \le M_R). \quad (18)$$

Once the CDF is known, the PMF of $N_{F(j)}$ can be easily calculated from (18):

$$pmf_{NF(j)}(k) = \begin{cases} cdf_{NF(j)}(k) & (k = 0) \\ cdf_{NF(j)}(k) - cdf_{NF(j)}(k-1) & (1 \le k \le M_R) \end{cases}. \quad (19)$$

The mean value of $N_{F(j)}$ (i.e., the average number of failed bit cells for the jth sorted column) is equal to:

$$E[N_{F(j)}] = \sum_{k=0}^{M_R} k \cdot pmf_{NF(j)}(k). \quad (20)$$

Based on (20), we can calculate the failure probability for the bit cells in the jth sorted column. To this end, we re-write $N_{F(j)}$ as:

$$N_{F(j)} = \sum_{i=1}^{M_R} \xi_{i(j)} \quad (21)$$

where the random variables $\{\xi_{i(j)}; i = 1,2,\ldots,M_R\}$ are used to model the failure events of the bit cells in the jth sorted column. They satisfy the following Bernoulli distribution:

$$p[\xi_{i(j)}] = \begin{cases} \alpha_{(j)} & [\xi_{i(j)} = 1] \\ 1 - \alpha_{(j)} & [\xi_{i(j)} = 0] \end{cases} \quad (22)$$

where $\xi_{i(j)} = 1$ indicates that the ith cell in the jth sorted column is failed. Eq. (21)-(22) are similar to (13)-(14) except that the random variables in (21)-(22) are associated with the failure events after the repair is made.

Comparing (13)-(14) and (21)-(22), we can have two important observations. First, the failure probability $\alpha_{(j)}$ in (22) is different from α in (13), since the repair process can substantially change the failure rate, as discussed at the beginning of this sub-section. Second, for different bit cells in the same column, they share the same Bernoulli distribution. In other words, while different bit cells in different columns correspond to different failure probabilities, all bit cells in the same column share the same failure rate. Hence, combining (20)-(22), we have:

$$E[N_{F(j)}] = \sum_{i=1}^{M_R} E[\xi_{i(j)}] = M_R \cdot \alpha_{(j)}. \quad (23)$$

Substituting (20) into (23) yields:

$$\alpha_{(j)} = \sum_{k=0}^{M_R} k \cdot pmf_{NF(j)}(k) \Big/ M_R. \quad (24)$$

From (22) and (24), we calculate the second-order moment of $\xi_{i(j)}$:

$$E[\xi_{i(j)}^2] = \alpha_{(j)} = \sum_{k=0}^{M_R} k \cdot pmf_{NF(j)}(k) \Big/ M_R. \quad (25)$$

Since the jth sorted column is used to store the jth bit of the numerical data, the second-order moment $E[\xi_{i(j)}^2]$ in (25) exactly equals the expected value $E(|\delta_j|^2)$ in (12):

$$E\left(|\delta_j|^2\right) = \sum_{k=0}^{M_R} k \cdot pmf_{NF(j)}(k) \Big/ M_R. \quad (26)$$

Next, we should further calculate the second-order statistics $E(|\delta_i|\cdot|\delta_j|)$ $(i \ne j)$ so that the noise power $E(\delta^2)$ in (12) can be fully estimated. For this purpose, we need to calculate the joint probability distribution for $N_{F(i)}$ and $N_{F(j)}$. Based on the theories of order statistics [14], the joint CDF of $N_{F(i)}$ and $N_{F(j)}$, where $0 \le i < j \le M_C$, can be expressed as (27) at the bottom of this page. Here, we only consider the cases for $i < j$, because $cdf_{NF(i)(j)}(k_i,k_j)$ is equal to $cdf_{NF(j)(i)}(k_j,k_i)$. Once the CDF is known, we can calculate the joint PMF by (28), and then the expected value:

$$E[N_{F(i)} \cdot N_{F(j)}] = \sum_{k_i=0}^{M_R} \sum_{k_j=0}^{M_R} k_i \cdot k_j \cdot pmf_{NF(i)(j)}(k_i, k_j) \quad (i \ne j). \quad (29)$$

On the other hand, substituting (21) into $E[N_{F(i)} \cdot N_{F(j)}]$ yields:

$$E[N_{F(i)} \cdot N_{F(j)}] = E\left[\sum_{m=1}^{M_R} \sum_{n=1}^{M_R} \xi_{m(i)} \cdot \xi_{n(j)}\right] \quad (i \ne j). \quad (30)$$

Remember that all bit cells in the same column share the same distribution. Hence, we have:

$$E[N_{F(i)} \cdot N_{F(j)}] = M_R^2 \cdot E[\xi_{m(i)} \cdot \xi_{n(j)}] \quad (i \ne j). \quad (31)$$

Combining (29) and (31) results in:

$$E[\xi_{m(i)} \cdot \xi_{n(j)}] = \frac{1}{M_R^2} \cdot \sum_{k_i=0}^{M_R} \sum_{k_j=0}^{M_R} k_i k_j \cdot pmf_{NF(i)(j)}(k_i, k_j) \quad (i \ne j). \quad (32)$$

The expected value $E[\xi_{m(i)} \cdot \xi_{n(j)}]$ in (32) exactly equals to $E(|\delta_i|\cdot|\delta_j|)$ in (12):

$$E[\delta_{(i)} \cdot \delta_{(j)}] = \frac{1}{M_R^2} \cdot \sum_{k_i=0}^{M_R} \sum_{k_j=0}^{M_R} k_i k_j \cdot pmf_{NF(i)(j)}(k_i, k_j) \quad (i \ne j). \quad (33)$$

Substituting (26) and (33) into (12), we can estimate the upper bound of the noise power $E(\delta^2)$ due to bit cell failure. It, in turn, facilitates us to estimate the SNR in (11) that provides a quantitative measure of the amount of information stored in a memory system.

$$cdf_{NF(i)(j)}(k_i, k_j) = \begin{cases} \displaystyle\sum_{n=j}^{M_C} \sum_{m=i}^{j} \frac{M_C!}{m!(n-m)!(M_C-n)!} \cdot cdf_{NF}^{m}(k_i) \cdot [cdf_{NF}(k_j) - cdf_{NF}(k_i)]^{n-m} \cdot [1 - cdf_{NF}(k_j)]^{M_C-n} & (0 \le k_i < k_j \le M_R) \\ cdf_{NF(i)(j)}(k_i, k_j) = cdf_{NF(j)}(k_j) & (0 \le k_j \le k_i \le M_R) \end{cases} \quad (27)$$

$$pmf_{NF(i)(j)}(k_i, k_j) = \begin{cases} cdf_{NF(i)(j)}(k_i, k_j) & (k_i = 0 \;\&\; k_j = 0) \\ cdf_{NF(i)(j)}(k_i, k_j) - cdf_{NF(i)(j)}(k_i - 1, k_j) & (k_i \ge 1 \;\&\; k_j = 0) \\ cdf_{NF(i)(j)}(k_i, k_j) - cdf_{NF(i)(j)}(k_i, k_j - 1) & (k_i = 0 \;\&\; k_j \ge 1) \\ cdf_{NF(i)(j)}(k_i, k_j) - cdf_{NF(i)(j)}(k_i - 1, k_j) - cdf_{NF(i)(j)}(k_i, k_j - 1) + cdf_{NF(i)(j)}(k_i - 1, k_j - 1) & (k_i \ge 1 \;\&\; k_j \ge 1) \end{cases} \quad (28)$$

Algorithm 1 summarizes the major steps of the aforementioned flow for information modeling. Starting from a set of given design parameters (e.g., array size, word length, etc.), Algorithm 1 first calculates the power of the quantization error $E(\varepsilon^2)$ by using (4). Next, it applies the analytical equations (26) and (33) to calculate the second-order statistics $E(|\delta_i|\cdot|\delta_j|)$ where $i,j \in \{1,2,\ldots,N\}$, and then the noise power $E(\delta^2)$ in (12). Finally, combining (4) and (12) yields the SNR in (11).

Algorithm 1: Information Modeling of MIMS

1. Start from a set of given design parameters for an SRAM array, including the number of rows M_R, the number of columns M_C, the word length N, and the bit cell failure probability α (without repairing).
2. Compute the power of the quantization error $E(\varepsilon^2)$ by using (4).
3. Calculate the functions $pmf_{NF}(k)$ and $cdf_{NF}(k)$ in (15)-(16), before the repair scheme is applied.
4. Calculate the functions $cdf_{NF(j)}(k)$ and $pmf_{NF(j)}(k)$ in (18)-(19) and the second-order moment $E(|\delta_j|^2)$ in (26) for $j \in \{1,2,\ldots,N\}$.
5. Calculate the functions $cdf_{NF(i)(j)}(k_i,k_j)$ and $pmf_{NF(i)(j)}(k_i,k_j)$ in (27)-(28) and the expected value $E(|\delta_i|\cdot|\delta_j|)$ in (33) for $i,j \in \{1,2,\ldots,N\}$ and $i \neq j$.
6. Estimate the upper bound of the noise power $E(\delta^2)$ due to bit cell failure by using (12).
7. Estimate the SNR in (11).

B. Design Optimization

Algorithm 1 provides an efficient way to estimate the SNR for a given memory system. As shown in [11], maximum-information storage can be achieved by maximizing the SNR in (11). Hence, SNR can be used as a quantitative measure of design optimality to compare different memory systems. In other words, given the statistical algorithm developed in Section 3.2A, we can explore a number of design parameters (e.g., the number of rows M_R, the number of columns M_C, the word length N, etc.) to maximize the information density of data storage. As will be demonstrated by an image processing example in Section 4, the optimal value of these design parameters strongly depends on the technology node (e.g., the device-level variability) and the environmental setup (e.g., the power supply voltage). Significant information loss will occur, if the memory system is not optimally designed.

4. NUMERICAL EXAMPLES

In this section, we demonstrate the efficacy of the proposed MIMS system by using an image processing example. Given a 6-T SRAM cell designed in a commercial 45nm CMOS process, we apply Monte Carlo analysis (i.e., importance sampling [4], [7]) to estimate the failure rate α of the SRAM cell due to process variations. Such a Monte Carlo analysis is repeatedly performed at different power supply voltages, since we are interested in the behavior of the SRAM with reduced supply voltage for low-power operations. Table 1 shows the failure rate α as a function of the power supply voltage V_{DD}.

Next, two different SRAM designs are implemented for testing and comparison purposes. First, a traditional memory array is created with M_R = 64 rows and 10% column redundancy. In other words, if the memory array has M_C columns in total, there are 10%×M_C redundant columns reserved for memory repair. Hence, the word length N is equal to 90%×M_C. During the repair process, we count the number of failed bit cells for each column. The first N columns with the minimal number of failed bit cells are used for data storage. However, among these N columns, we do not distinguish the levels of importance for different bits. These N columns are randomly mapped to the N bits of the numerical data.

On the other hand, we consider the optimal repair scheme for the proposed MIMS system. Similar to the previous case, the memory array has M_R = 64 rows. For a given number of columns (i.e., M_C), Algorithm 1 is applied to estimate the SNR in (11) for different values of the word length N. The optimal word length N_{OPT} is then selected to achieve the maximal SNR. During the repair process, we count the number of failed bit cells for each column. The first N_{OPT} columns with the minimal number of failed bit cells are used for data storage. In addition, these N_{OPT} columns are optimally mapped to the different bits of the numerical data. The important bits are stored in the columns with a small number of failed bit cells.

Table 1. Failure rate of a 6-T 45nm SRAM cell as a function of the power supply voltage V_{DD}

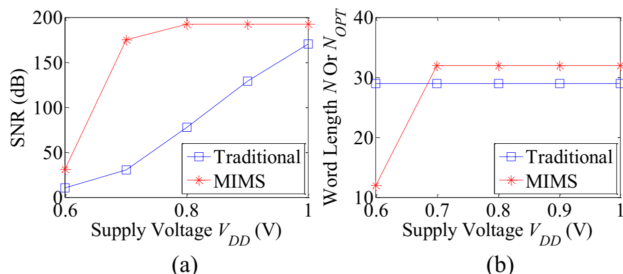
V_{DD} (V)	0.6	0.7	0.8	0.9	1.0
Failure Rate	2.7×10^{-2}	1.3×10^{-3}	6.2×10^{-5}	3.2×10^{-6}	2.6×10^{-7}

4.1 Bit Cell Repair

Figure 3(a) shows the SNR values, as the supply voltage V_{DD} varies from 0.6 V to 1.0 V. In these experiments, the SRAM array size is set to 64 (rows) by 32 (columns). Note that the proposed MIMS system substantially improves the SNR by more than 100 dB over the traditional SRAM design. Such a significant improvement is achieved due to two reasons. First, MIMS optimally maps the columns of the SRAM array to different bits of the numerical data by considering their levels of importance. Second, MIMS optimally determines the word length N_{OPT} that maximizes the SNR. As shown in Figure 3(b), while the traditional SRAM design uses a fixed word length (i.e., N = 90%×M_C), the optimal word length N_{OPT} determined by MIMS varies with the supply voltage V_{DD}. For instance, if V_{DD} is low and, hence, the failure rate of bit cells is large, MIMS selects a small word length N_{OPT} so that a large number of SRAM columns can be reserved as redundant columns for bit cell repair. On the other hand, if V_{DD} is high and the bit cells are unlikely to fail, MIMS increases the word length N_{OPT} to minimize the quantization error which becomes the dominant noise power at a high V_{DD}.

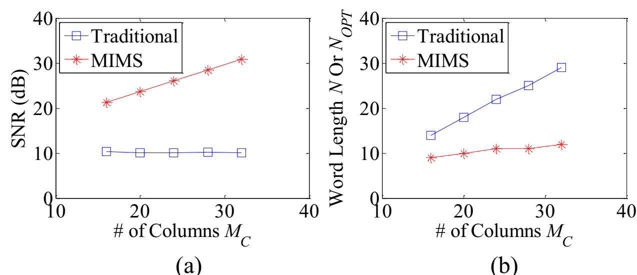
Next, we further compare the two SRAM designs where the memory array is set to different sizes. Figure 4(a) shows the SNR values, as the number of SRAM columns M_C varies from 16 to 32. In these experiments, the number of rows M_R is set to 64 and the power supply voltage V_{DD} is set to 0.6 V. Since V_{DD} is low, each individual bit cell has a large failure rate. Studying Figure 4(a), we would find that the proposed MIMS design improves the SNR by up to 20 dB over the traditional SRAM system. Figure 4(b) further plots the word length N (traditional) or N_{OPT} (MIMS) as a function of M_C. Note that the optimal word length N_{OPT} of MIMS is almost unchanged. It implies that as M_C increases, more and more columns are used by MIMS for bit cell repair in order to minimize the noise power due to bit cell failure, thereby improving the SNR as shown in Figure 4(a). On the other hand, the traditional SRAM design increases the word length N with M_C (i.e., N = 90%×M_C). While such a strategy reduces the quantization noise, it does not help to repair the bit cell failure which contributes to the dominant information loss in this example. For this reason, the traditional SRAM design fails to improve the SNR, even if M_C increases and a large number of

SRAM columns are available, as shown in Figure 4(a).



(a) (b)

Figure 3. Compare the traditional SRAM design and the proposed MIMS system at different supply voltages (M_R = 64 and M_C = 32): (a) SNR, and (b) word length N (traditional) or N_{OPT} (MIMS).



(a) (b)

Figure 4. Compare the traditional SRAM design and the proposed MIMS system with different numbers of columns (M_R = 64 and V_{DD} = 0.6 V): (a) SNR, and (b) word length N (traditional) or N_{OPT} (MIMS).

4.2 Image Processing Application

(a) (b) (c)

Figure 5. (a) original image, (b) image stored by the traditional SRAM design (SNR = 6.6 dB), and (c) image stored by the proposed MIMS system (SNR = 26.9 dB).

Shown in Figure 5(a) is a benchmark example that has been widely used to test various image processing algorithms [15]. The image size is 512×512. It is in BMP format where each pixel is a numerical number. For testing and comparison, we store this BMP image in two different memory systems: (1) the traditional SRAM and (2) the proposed MIMS. For both memory designs, one SRAM array contains 64 rows and 32 columns, and it operates at V_{DD} = 0.6 V. As shown in Figure 5(b), the traditional SRAM design results in a low SNR (i.e., 6.6 dB), particularly due to the large failure rate of bit cells at the low power supply voltage. On the other hand, the proposed MIMS system is able to significantly improve the SNR by more than 20 dB, as shown in Figure 5(c). In this example, even though the bit cells are not robustly designed for V_{DD} = 0.6 V, MIMS is able to repair the important bits and, hence, provide a high SNR (i.e., 26.9 dB).

5. CONCLUSIONS

In this paper, we propose a new bit cell repair scheme for maximum-information memory system (MIMS). Our goal is to maximize the information density for data storage. The proposed technique results in an optimal SRAM design with maximal SNR for a number of application-specific cases such as signal processing and numerical computation. To optimally design the proposed MIMS system, a new statistical algorithm is developed to efficiently estimate the SNR and then optimize the design parameters (e.g., the number of rows M_R, the number of columns M_C, the word length N, etc.) to achieve maximum-information storage. Our experimental results demonstrate that even with a traditional 6-T SRAM cell designed in a commercial 45nm CMOS process, the proposed MIMS system can successfully operate at an extremely low power supply voltage (i.e., 0.6 V). It achieves more than 20 dB improvement in SNR compared to the traditional SRAM design.

6. ACKNOWLEDGEMENTS

The author acknowledges the support of the C2S2 Focus Center, one of six research centers funded under the Focus Center Research Program (FCRP), a Semiconductor Research Corporation entity.

7. REFERENCES

[1] B. Calhoun, Y. Cao, X. Li, K. Mai, L. Pileggi, R. Rutenbar and K. Shepard, "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] A. Bhavanagarwala, X. Tang and J. Meindl, "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," *IEEE JSSC*, vol. 36, no. 4, pp. 658-665, Apr. 2001.

[3] S. Mukhopadhyay, K. Kim, H. Mahmoodi and K. Roy, "Design of a process variation tolerant self-repairing SRAM for yield enhancement in nanoscaled CMOS," *IEEE JSSC*, vol. 42, no. 6, pp. 1370-1382, Jun. 2007.

[4] 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," *IEEE DAC*, pp. 69-72, 2006.

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

[6] C. Gu and J. Roychowdhury, "An efficient, fully nonlinear, variability-aware non-Monte-Carlo yield estimation procedure with applications to SRAM cells and ring oscillators," *IEEE ASPDAC*, pp 754-761, 2008.

[7] L. Dolecek, M. Qazi, D. Shah and A. Chandrakasan, "Breaking the simulation barrier: SRAM evaluation through norm minimization," *IEEE ICCAD*, pp. 322-329, 2008.

[8] J. Wang, S. Yaldiz, X. Li and L. Pileggi, "SRAM parametric failure analysis," *IEEE DAC*, pp. 496-501, 2009.

[9] J. Jaffari and M. Anis, "Adaptive sampling for efficient failure probability analysis of SRAM cells," *IEEE ICCAD*, pp. 623-630, 2009.

[10] A. Bansal, R. Singh, R. Kanj, S. Mukhopadhyay, J. Lee, E. Acar, A. Singhee, K. Kim, C. Chuang, S. Nassif, F. Heng and K. Das, "Yield estimation of SRAM circuits using virtual SRAM fab," *IEEE ICCAD*, pp. 631-636, 2009.

[11] X. Li, "Maximum-information storage system: concept, implementation and application," *IEEE ICCAD*, pp. 39-46, 2010.

[12] A. Oppenheim, R. Schafer and J. Buck, *Discrete-Time Signal Processing*, 1999.

[13] A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 2001.

[14] H. David and H. Nagaraja, *Order Statistics*, Wiley-Interscience, 2003.

[15] R. Gonzalez and R. Woods, *Digital Image Processing*, Prentice Hall, 2007.